

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ТЕПЛОЕНЕРГЕТИЧНИЙ ФАКУЛЬТЕТ

Кафедра автоматизації проектування енергетичних процесів і систем

«На правах рукопису»
УДК 621.43.053:632.18

«До захисту допущено»

Завідувач кафедри

_____ О.В. Коваль
(підпис) (ініціали, прізвище)

“ ” _____ 2019 р.

**Магістерська дисертація
на здобуття ступеню магістра**

зі спеціальності 121 – Інженерія програмного забезпечення
(код і назва спеціальності)

спеціалізації Інженерія програмного забезпечення розподілених систем

На тему: Система розрахунку екологічних платежів мережі АЗС

Виконав: студент VI курсу, групи ТВ-82мп

_____ Олексій Артур Олегович

(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник доцент, к.т.н Гагарін О. О.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІГОРЯ СІКОРСЬКОГО»**

Факультет (інститут) Теплоенергетичний
(повна назва)

Кафедра Автоматизації проектування енергетичних процесів і систем
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність 121 Інженерія програмного забезпечення
(код і назва)

спеціалізація Інженерія програмного забезпечення розподілених систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис)

(ініціали, прізвище)

«__» _____ 20__ р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Олексій Артур Олегович

(прізвище, ім'я, по батькові)

1. Тема дисертації Система розрахунку екологічних платежів мережі АЗС

науковий керівник дисертації доцент, к.т.н Гагарін О. О.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «04» листопада 2019 р. №3813-с

2. Строк подання студентом дисертації 10 грудня 2019

3. Об'єкт дослідження Обчислення викидів та платежів мереж

4. Предмет дослідження Програмне забезпечення для обчислення, перегляду, аналізу викидів мережі АЗС та супроводу системи

5. Перелік завдань, які потрібно розробити

5.2. Дослідження методики обчислення викидів

5.1. Обчислення викидів АЗС за різними критеріями

5.2. Обчислення екологічних виплат за викид

- 5.3. Доступ до даних для перегляду, аналізу та редагування
- 5.4. Можливість надання даних для заповнення податкової декларації
- 5.5. Можливість супроводу системи

6. Орієнтовний перелік ілюстративного (графічного) матеріалу:

6.1. Презентація PowerPoint відповідно до теми дисертації.

7. Орієнтовний перелік публікацій

7.1. Доповідь на науково-технічній конференції з публікацією тез

8. Дата видачі завдання «____»_____201__р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Затвердження теми роботи	до 07.05.2019	
2	Вивчення та аналіз задачі. Проведення дослідження по вибраній темі	до 02.09.2019	
3	Розробка архітектури та загальної структури системи	до 30.09.2019	
4	Програмна реалізація системи	до 22.10.2019	
5	Захист програмного продукту	до 25.10.2019	
6	Оформлення пояснювальної записки	до 9.12.2019	
7	Передзахист	до 22.11.2019	
8	Захист	з 17.12.2018 р.	

Студент

(підпис)

Олексій А.О.

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

Гагарін О.О

(ініціали, прізвище)

РЕФЕРАТ

Тема: “Система розрахунку екологічних платежів мережі АЗС”

Виконав: магістрант групи ТВ-82мп Олексій Артур Олегович

Кількість сторінок записки 91

Актуальність. Важливим пунктом охорони навколишнього середовища є збір екологічного податку за забруднення навколишнього середовища, в тому числі і з мереж АЗС. Перед підприємством постає ряд зобов’язань, за невиконання яких можуть виникати подальші санкції у вигляді додаткових штрафів, зупинки роботи підприємств та відповідальності в залежності від ступеня тяжкості порушення. Кожного кварталу, підприємство зобов’язане здавати додаток до податкової декларації, в якому будуть вказані об’єми викидів, що здійснює підприємство та виплат за них. Також, регулюються в залежності від класу підприємства по забрудненням та об’ємам викидів. Для рішення подібних задач необхідне зручне програмне забезпечення, що вирішуватиме ряд завдань, що стоять перед підприємством.

Ціль та завдання. Ціллю є розробка програмного продукту, що матиме можливість здійснювати обчислення викидів та виплат мережі АЗС, аналізувати та представляти дані, підтримувати склад сховища даних

Для досягнення поставленої цілі, необхідно вирішити наступні завдання:

- дослідження вимог та методики обчислення викидів та виплат;
- вивчення можливих програмних засобів реалізації програмного продукту;
- формування вимог до програмного продукту;
- програмна реалізація проекту;
- розробка стратап-проекту;

Об’єкт дослідження. Програмне забезпечення для обчислення викидів та екологічних виплат мережі АЗС.

Предмет дослідження. Предметом дослідження є вимоги до АЗС щодо екологічних викидів і виплат, програмні засоби розробки програмного забезпечення та його реалізації

Наукова новизна. Наукова новизна роботи полягає в дослідженні вимог та методів обчислення екологічних викидів та виплат за них.

Практична цінність. Практична цінність роботи полягає у аналізі методів та засобів для обчислень екологічних виплат, розробленні мобільного додатку як прикладу їх використання.

Публікації.

Олексій А. О. Система оцінювання екологічних збитків мережі / Олексій : Матеріали XVII міжнародної науково-практичної студентської конференції, 23-26 квітня 2019 року, Київ, Україна : матеріали. – К. : НТУУ «КПІ ім. Ігоря Сікорського», 2019. – С. 96

Ключові слова: iOS, екологічні викиди, екологічні виплати, АЗС.

ABSTRACT

Subject: "System for calculating the environmental payments of the gas station network"

Completed: undergraduate of the group TV-82mp Oleksiy Arthur Olegovich

Number of Pages: 90

Relevance. An important point of environmental protection is the collection of an environmental tax for environmental pollution, including from gas station networks. The company has a number of obligations, for the failure of which may result in further sanctions in the form of additional fines, shutdown of enterprises and liability depending on the severity of the violation. Every quarter, the company is obliged to submit an addition to the tax return, which will indicate the amount of emissions, the company makes and payments on them. Also, they are regulated depending on the class of the enterprise in terms of pollution and emissions. To solve such problems, you need convenient software, to solve a number of problems facing the enterprise.

Goal and tasks. The goal is to develop a software product that will calculate the emissions and payments of the gas station network, analyze and present data, support the composition of the data warehouse

To achieve this goal it is necessary to solve the following tasks:

- study of requirements and methods for calculating emissions and payments;
- study of possible software tools for implementing a software product;
- formation of requirements for a software product;
- software implementation of the project;
- development of a startup project;

Object of study. Software for calculating emissions and environmental benefits of a gas station network.

Subject of study. The subject of the study is the requirements for gas stations for environmental emissions and payments, software tools for software development and its implementation

Scientific novelty. The scientific novelty of the work is to study the requirements and methods for calculating environmental emissions and payments for them.

Practical value. The practical value of the work lies in the analysis of methods and tools for computing environmental payments, the development of a mobile application as an example of their use.

Publications. Oleksiy A.O. Network environmental damage assessment system / Materials of the XVIII International Scientific and Practical Student Conference, April 23-26, 2019, Kiev, Ukraine: materials. - M .: NTUU "KPI them. Igor Sikorsky ", 2019. - P. 96.

Keywords: iOS, environmental emissions, environmental payments, gas stations.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ ..	10
ВСТУП.....	12
1 ЗАДАЧА РОЗРОБКИ ПРОГРАМНОГО ЗАСОБУ ДЛЯ РОЗРАХУНКУ ЕКОЛОГІЧНИХ ПЛАТЕЖІВ	13
1.1 Мета роботи	13
Висновки до розділу 1	13
2 АНАЛІЗ ПРОБЛЕМИ СТВОРЕННЯ	14
2.1 Основні вимги.....	14
2.2 Теоретичні дані.....	15
2.3 Предметна область	19
Висновки до розділу 2	20
3 ВИКОРИСТАНІ ПРОГРАМНІ ЗАСОБИ	21
3.1 Середовищ розробки.....	21
3.2 Мови програмування	22
3.3 Microsoft Azure	24
3.4 Firebase	26
3.5 Бібліотека Charts.....	32
3.5 Середовище емуляції	34
Висновки до розділу 3	35
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	36
4.1 Програмна реалізація	36
4.2 Збереження даних.....	44
4.3 Архітектурний шаблон	46
4.4 Тестування	48
5 РОБОТА КОРИСТУВАЧА З ПРОГРАМОЮ	55
5.1 Інсталяція програми.....	55
5.2 Вхід в систему	55
5.3 Обчислення викидів	57
5.3 Робота з даними.....	58
5.4 Супровід додатку	62

Висновки до розділу 5	66
6 РОЗРОБКА СТАРТАП-ПРОЕКТУ	67
6.1 Опис ідеї проекту	67
6.2 Технологічний аудит ідеї проекту	69
6.3 Аналіз ринкових можливостей запуску стартап-проекту	70
6.4 Розроблення ринкової стратегії проекту	76
6.5 Розробка мартекингової програми	79
Висновки до розділу 6	83
ВИСНОВКИ.....	84
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	85
ДОДАТОК А.....	88

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

A3C – автозапрадна станція

БД – база даних

V – об'єм речовини

ПЗ – програмне забезпечення

SDK (software Development Kit) – набір програмних інструментів для розробки програмного забезпечення

API (application programming interface) – опис методів, якими одне програмне забезпечення може взаємодіяти з іншим.

СУБД – система управління базами даних

HTTP (Hypertext Transfer Protocol) - протоколом прикладного рівня для передачі даних.

SaaS - однакова із форм облачних обчислень, модель обслуговування, при якій підписникам пропонують готову прикладну програму, що повністю обслуговується провайдером.

IaaS - одна з моделей обслуговування у хмарних обчисленнях, за якими користувачі пропонують за підпискою фундаментальні інформаційно-технологічні ресурси - віртуальні сервери з заданою потужністю та операційною системою

PaaS - модель надання хмарних обчислень, за якими користувач отримує доступ до використаних інформаційно-технологічних платформ: операційні системи, система управління базою даних, що використовуються програмним забезпеченням, засоби розробки та тестування, розміщені в хмарного провайдері.

SQL(structured query language) - декларативний мова програмування, яка застосовується для створення, модифікації та управління даними в реляційній базі даних, керована відповідною системою управління базами даних.

NoSQL(not only SQL) - термін, відокремлюючи ряд підходів, створених на реалізацію системи управління БД, які відрізняються від звичних реляційних БД

іншими підходами та моделями побудови та взаємодії.

PNG, JPEG – формати зображення

AppStore – онлайн-магазин додатків для наступних платформ: macOS, iOS, iPadOS, watchOS та tvOS.

ВСТУП

Одним з головних завдань по охороні навколишнього середовища займає збір екологічного податку за забруднення навколишнього середовищ, в тому числі і з мереж АЗС. Перед підприємством постає ряд зобов'язань, за невиконання яких можуть виникати подальші санкції у вигляді долаткових штрафів, зупинки роботи підприємств та відповідальності в зв'язності від ступеня тяжкості порушення. Кожного кварталу, підприємство зобов'язане здавати додаток до податкової декларації, в якому будуть вказані об'єми викидів, що здійснює підприємство та виплат за них. Станція АЗС здійснюють різні типи викидів, кожен з яких можливо порахувати. Основними типами забруднення є викиди при заповненні цистерни, викиди при заправці автомобіля та викиди при збереженні палива. Викиди нормуються, та за порушення допустимих норм також можливі санкції щодо мережі. Закон намагається регулювати та штрафувати подібні підприємства, але для контролю за ними потрібно використовувати сучасні та ефективні інструменти.

Для вирішення поставлених завдань необхідно використовувати багатофункціональне ПЗ, що вирішує значну кількість задач. Наявне програмне забезпечення існує, але воно застаріле та його функціонал не відповідає сучасним вимогам. Розробка нового, зручнішого програмного продукту з використанням сучасних технологій, новим інтерфейсом та повним функціоналом зможе зацікавити багатьох зацікавлених покупців та потіснити конкурентів.

В даній роботі було досліджено можливості хмарної СУБД Cloud Firestore, розроблено модулі по обчисленню викидів, представленню та аналізу даних, модуль підтримки наявних даних системи.

1 ЗАДАЧА РОЗРОБКИ ПРОГРАМНОГО ЗАСОБУ ДЛЯ РОЗРАХУНКУ ЕКОЛОГІЧНИХ ПЛАТЕЖІВ

Метою роботи є розгляд проблеми завдання та вимог щодо можливої програмної реалізації.

1.1 Мета роботи

Метою розробки є реалізація програмного засобу, що надає можливість обчислення викидів, аналізу даних та обчислення екологічних виплат АЗС. Програмний продукт повинен вирішувати ряд задач, що стоять перед ним.

Потрібен зручний, інтуїтивно зрозумілий інтерфейс, що дозволить просто та ефективно вирішувати поставлені задачі. Правильне подання інформації, що не тільки вирішуватиме ряд поставлених завдань, але й допоможе користувачеві розібратись в самій системі та її даних. Розроблена система повинна бути гнучкою та підтримуваною, передбачати різні сценарії роботи з нею.

Необхідними можливостями, які має забезпечувати програма, є:

- можливість введення та обчислення вхідних даних;
- представлення даних в графічному та текстовому форматах;
- аналіз отриманих даних;
- можливість керувати даними, пов'язаними з мережею АЗС;
- подання даних для заповнення додатку до податкової деклар.

Висновки до розділу 1

В даному розділі було розглянуто мету та основні вимоги до програмного продукту, сформовано предметну область дослідження.

2 АНАЛІЗ ПРОБЛЕМИ СТВОРЕННЯ

В розділі будуть розглянуті основні вимоги до екологічних викидів та виплат підприємства. Також будуть вказані формули, що використовувались при обчисленнях та предметна область, що опише основні характеристики, викиди та основні процеси.

2.1 Основні вимги

Виплати за викиди до оточуючого середовища є важливою законодавчого механізму охорони та зменшення шкоди від експлуатації шкідливих для оточуючого середовища речовин. Розмір та критерії оподаткування виизначені відповідними нормативними документами[2].

Загалом, встановлено три основні види забруднення навколишнього середовища, що поділяються на:

- забруднення атмосфери зі стаціонарних джерел;
- викиди відходів та небезпечних речовин у водні об'єкти;
- збереження відходів на своїй території.

Характер викидів до навколишнього середовища дуже різноманітний і не завжди піддається кількісному обліку. Дуже складно охарактеризувати та вести облік характеристики викидів. Основними типами втрат від здійснення викидів мають як соціальну складову, так і економічну. Соціальна складова впливає в наслідок небезпечного впливу на людське здоров'я, оскільки спричиняє ряд захворювань, спостерігається скорочення терміну життя та пониження продуктивності праці. Економічними наслідками є зменшення кількості виготовленого продукту та зменшення погіршення загальної результативності підприємства.

Сучасна політика держав в галузі охорони оточуючого середовища від забруднення будується на принципі “забруднювач платить”. Цей принцип передбачає, що той, хто спричинив шкоду екологічній обстановці, повинен понести

відповідальність та здійснити виплати за свою діяльність. Принцип був сформульований у кінці 70-х років XX сторіччя. За чотири десятиліття, принцип був імплементований в якості заноїв та законодачих актів в багатьох країнах світу. В нашій країні, цей принцип запровадили 1991 році. В Законі України “Про охорону навколишнього природного середовища”, є стаття 44, що передбачає саме поняття плати за забруднення оточуючого середовища. Постановою Кабінету Міністрів України від 13 січня 1992 року, було прийнято механізм, що визначав вимоги та принцип оподаткування за забруднення оточуючого середовища. З 1999 року, постановою Кабінету Міністрів України щорічно регулюється розмір оподаткування підприємств, здійснюючих шкідливий вплив.

Платежі за забруднення навколишнього середовища є складовою частиною фінансового механізму охорони довкілля та раціонального використання природних ресурсів. основу платежів становлять нормативи плати за забруднення навколишнього середовища. Отже, створено механізми покарання та регулювання діяльності підприємства, що здійснює забруднення.

2.2 Теоретичні дані

Є мережа АЗС. До складу кожної АЗС входить визначена кількість резервуарів. Для кожного з них визначений тип бензину: А-80, А-92, А-95, А-98 та дизельного паливо.

Таким чином, основний викид вуглеводнів в атмосферу вираховується із норм природної втрати палива, яка враховує втрати нафтопродуктів при прийомі, збереженні, відпуску споживачам.

Нижче наведені теоретичні дані для розрахунку викидів в атмосферу по типах забруднення.

При зливі рідини під атмосферним тиском викиди в атмосферу проходять через «зворотній видих», за рахунок витіснення із цистерни повітря, насиченої парами зливної рідини. Їх можна обчислити по формулі 2.1:

$$n_i = 1,2 \times \sum Q \times k_i \times x_i \frac{M_i}{273 + t_{ob}} \quad (2.1)$$

1,2 – коефіцієнт, що складає 10% від величини «великого видиху»;

Q – річний об'єм зливаючої із цистерни рідини, м³/рік;

x_i - мольна частина i -ої речовини в рідині;

k_i - константа рівноваги, яка вираховується за формулою:

$$k_i = \frac{P_i}{P_a} \quad (2.2)$$

P_i - парціальний тиск пари i -го компонента мм рт. ст. при $t_{ob} = 100^\circ\text{C}$;

P_a - атмосферний тиск;

t_{ob} - середньорічна температура навколишнього повітря;

$\sum Q$ - сумарна об'ємна маса рідини, що наливається в резервуари чи групу резервуарів у вказаний період;

M_i - молекулярна маса i -ої речовини, кг/моль, молекулярна маса бензинів по формулі Воїнова:

$$M_i = 0,3t_{cp} + 0,001 \times t_{cp}^2 \quad (2.3)$$

де середня температура кипіння вуглеводневих рідин в результаті при тиску 760 мм рт.ст. складає:

$$t_{cp} = 0,5 \times (t_{cp} + t_{cp}^2) \quad (2.4)$$

де:

t_1 і t_2 - температури рідин при закипанні і в кінці закипання, $^\circ\text{C}$;

x_i - мольна частина i -ої речовини в рідині;

Кількість вуглеводів, що викидаються в атмосферу при заправці автомобіля за годину можна підрахувати згідно з «Нормами природної втрати при відпуску в транспортні засоби», формули 2.5-2.7[3]:

$$G = (n_1 \times G_h + n_2 \times G_w) \quad (2.5)$$

де:

$n_1=0,56$ – норма природної втрати нафтопродуктів в весняно-літній період, кг/т;

$n_2=0,48$ - норма природної втрати нафтопродуктів в осінньо-зимовий період, кг/т ;

G_h, G_w - кількість нафтопродуктів, прийнятих в резервуари у весняно-літній та осінньо-зимовий періоди.

Для дизельного палива - $n_1=0,02$ кг/т і $n_2=0,02$ кг/т

Річний викид вуглеводнів складає:

$$G = [V_b \times (0,48 + 0,56) + V_d \times 0,04] \times 10^{-6} \quad (2.6)$$

Кількість заправок за годину при часі заправки 250 за добу складе близько 11, при цьому річна кількість складе приблизно 96360 заправки.

Тоді усереднена потужність викиду при заправці автомобіля складе:

$$m = \frac{G \times 10^6}{k_a \times O \times 3600} \quad (2.7)$$

де:

k_a - кількість заправленого автотранспорту за добу;

O - середня кількість днів відпрацьованих в рік.

При зберіганні рідин в ємностях з'єднаних з атмосферою дихальними

клапанами викиди проходять при витісненні в атмосферу повітря насиченого їх парами, наступні формули:

Масову кількість шкідливих викидів парів вуглеводнів можна оцінити по наступній формулі[4]:

$$N_i = 13,4 \sum Q k_i x_i \frac{M_i}{273 + t_{ob}} K_2 K_3 \quad (2.9)$$

де:

13,4 – коефіцієнт, що враховує величину викидів від «великих дихань» при заповненні резервуарів водою, а також при відкачці рідини із резервуарів від «зворотнього видиху»;

$K_2=1,25$ – коефіцієнт, що враховує величину викидів від «малих дихань» резервуара, котрі проходять при випаровуванні рідин від нагріву атмосферним повітрям, чи від сонячної радіації;

$K_3=1,1$ – коефіцієнт, що враховує технічне обладнання по зменшенню витоків.

Потужність викиду з АЗС (кг/сек) чи (г/сек)

$$N_i = 0,037 m P_i V_{nr} y_i \sqrt{\frac{M_i}{(273 + t_{ob})}} \quad (2.10)$$

де:

$m = 0,1$ – приймається для горючих речовин;

P_i - парціальний тиск пари і-го компонента;

y_i - мольна частина речовини у паро газовій фазі

$$y_i = \left(\frac{P}{P_a} \right) \times x_i \quad (2.11)$$

де:

V_{nr} - об'єм парогазової фази, мЗ;

$$V_{nr} = V_{pes}(1 - g) \quad (2.12)$$

де:

V_{pes} - сумарний об'єм резервуарів;

g – коефіцієнт заповнення резервуарів;

Сума збору за забруднення атмосфери шкідливими речовинами обчислюється за формулою:

$$P = \sum_{i=1}^P M_{lim} + H_{bi} + K_{nas} + K_f \quad (2.13)$$

де:

M_{lim} - обсяг викинутої i -ої речовини;

H_{bi} - норматив за тону в гривнях;

K_{nas} - коефіцієнт чисельності жителів населеного пункту;

K_f - коефіцієнт народногосподарського значення населеного пункту.

2.3 Предметна область

Викиди зі станції АЗС мають місце при випаровуванні паливних випарів. Кожна АЗС має певну кількість цистерн визначеної моделі, кожна зі своїм об'ємом та пропускною здатністю паливних випарів.

Загалом, типи викидів з автозаправної станції можна класифікувати за наступними:

- викиди при заповненні палива в цистерну;
- викиди при заправці автомобіля;

— викиди при збереженні автомобіля.

Розглядаючи склад викидів палива, ми розглядаємо суміш вуглеводнів, що складається з наступних речовин:

- вуглеводні C1-C10;
- вуглеводні C2-C5;
- етил-бензол
- бензол;
- ксилол;
- толуол.

Типи викидів поділяються на “малі” та “великі” викиди. “Малі” викиди мають місце при зміні температури навколишнього середовища. Коли денна температура збільшується, має місце процес нагрівання парогазової фази в середині цистерни. Активізується процес випаровування рідини, що збільшує тиск в цистерні. При збільшенні тиску та досягнення його критичної точки призводить до виходу надлишкової частини парогазової фази та її викиду в атмосферу. Результат залежить від кількості рідини в цистерні.

”Великі видихи” відбуваються при заповненні резервуару новим паливом. При надходженні нової рідини до цистерни, збільшується тиск, в наслідок чого зменшується об’єм парогазового простору. Надлишкові випари виходять через дихальні клапани та надходять в атмосферу. Так відбувається процес ”великого видиху”. Значення викиду залежить від об’єму рідини, що надходить до резервуару при його заповненні та кліматичної зони.

Висновки до розділу 2

Було досліджено основні проблеми та вимоги щодо програмного продукту. Розібрано теоретичні дані та обрано формули для реалізації програмного продукту. Також, була розглянута предметна область та сформовано основні вимоги до мережі АЗС та складу їх станцій.

3 ВИКОРИСТАНІ ПРОГРАМНІ ЗАСОБИ

В даному розділі буде розглянуто основні програмні засоби, використані при розробці програмного продукту, а саме мову програмування Swift, інтегроване середовище розробки Xcode. Детальний опис та обґрунтування вибору даних засобів допоможуть зрозуміти переваги обраних програмних засобів.

3.1 Середовищ розробки

Xcode - це інтегроване середовище розробки, доступне для роботи на операційних системах macOS, що містить набір інструментів, створених компанією Apple для розробки програмного забезпечення для наступних платформ: macOS, iOS, iPadOS, watchOS та tvOS. Вперше випущене у 2003 році, Xcode доступне для завантаження в Mac App Store безкоштовно.

Середовище розробки підтримує код наступних мов програмування: C, C++, Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, ResEdit(та Swift та безліч програмних модулів, включаючи на не обмежуючись до Cocoa, Carbon та Java.

Xcode може компілювати великі за обсягом бінарні файли, що містять код для декількох архітектур з виконуваним форматом Mach-O. Вони називаються універсальними бінарними файлами, які дозволяють запускати програмне забезпечення як на платформах PowerPC, так і на базі Intel (x86) і які можуть включати як 32-бітний, так і 64-бітний код для обох архітектур. Використовуючи SDK iOS, Xcode також може використовуватися для компіляції та налагодження програм для iOS, які працюють на процесорах архітектури ARM.

Переваги:

- безоплатна установка та інтеграція усіх інструментів;
- наявність зручної емуляції додатків;
- найкраще середовище для моделювання користувацького інтерфейсу;

- підтримує повний цикл розробки додатку.

Недоліки:

- середовище орієнтоване на розробку під платформи компанії Apple.

AppCode - це інтегрована середовище розробки для розробки додатків з допомогою наступних мов: Swift, Objective-C, C, C ++ та JavaScript, побудована на платформі IntelliJ IDEA JetBrains. Перша загальнодоступна версія попереднього попереднього перегляду AppCode стала доступною у квітні 2011 року. Останній стабільний випуск показаний поруч зі стабільним випуском та доступний на офіційному веб-сайті JetBrains. AppCode побудований на платформі IntelliJ IDEA, яка написана на Java та Kotlin. Користувачі можуть розширити свої можливості, встановивши плагіни, створені для платформи IntelliJ, а також вони можуть писати власні плагіни.

Переваги:

- більше можливостей для рефакторингу коду;
- більша швидкість компіляції.

Недоліки:

- обмеженість ресурсів для розробки нативних додатків.

3.2 Мови програмування

Swift – відкрита, мультипарадигмова мова програмування загального призначення. Вона була розроблена та випущена в 2014 році компанією Apple для розробників iOS та macOS. Працює з фреймворками Cocoa та Cocoa Touch і сумісна з кодовою базою Apple. Протягом більш ніж 30 років, основною мовою програмування для продуктів компанії була мова Objective-C. За цей час вона морально застаріла та не задовольняла усіх потреб розробників. Тож на початку 2010 року, було прийнято рішення розробити нову мову програмування з простішим синтаксисом та більшою стійкістю до помилок в порівнянні зі своїм попередником.

Переваги:

- замість вказівників використовуються не вказівники, а типи даних, що обробляються компілятором;
- зручний синтаксис, простий для розуміння;
- наявність переважання операторів, узагальнене програмування та визначаємі найменування;
- сумісність з кодом, написаним на мовах C та Objective-C;
- менша кількість потенційних помилок, завдяки синтаксису та побудові мови;
- більша швидкість в порвнянні з іншими мовами програмування (рисунок 3.1).

Недоліки:

- відсутність стабільності ABI;
- менша швидкість компіляції;
- відсутність імпорту C++.

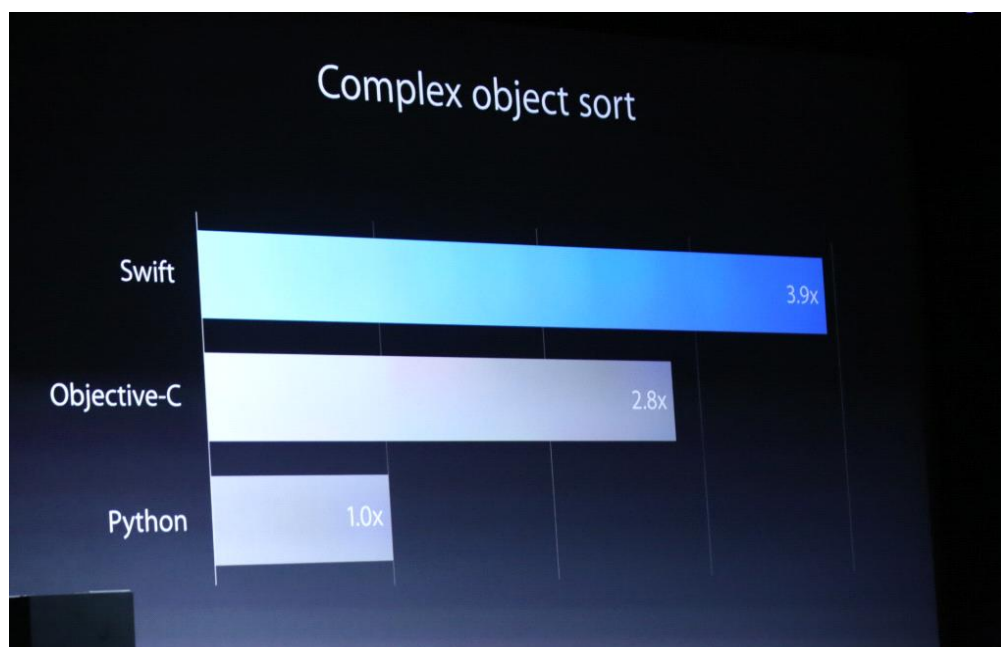


Рисунок 3.1 — Швидкість сортування комплексних об'єктів

Objective-C - це об'єктно-орієнтована мова програмування, яка додає повідомлення в стилі Smalltalk до мови програмування C. Це була основна мова

програмування, що підтримується Apple для операційних систем macOS, iOS та iPadOS та їх відповідних інтерфейсів прикладного програмування Cocoa та Cocoa Touch до введення Swift у 2014 році.

Objective-C - це тонкий шар на вершині мови програмування C і є "суворим набором" C, що означає, що можна компілювати будь-яку програму C за допомогою компілятора Objective-C і вільно включати код мови C у межах класу Objective-C. Objective-C отримує свій об'єктний синтаксис від Smalltalk. Усі синтаксиси для не об'єктно-орієнтованих операцій (включаючи примітивні змінні, попередню обробку, вирази, декларації функцій та виклики функцій) ідентичні тим, що мають C, тоді як синтаксис для об'єктно-орієнтованих функцій - це реалізація Smalltalk - стиль обміну повідомленнями.

Переваги:

- значна документація та спільнота;
- більша стабільність в порівнянні з іншими мовами;
- простіше використання приватних API;
- значна сумісність з мовою C++.

Недоліки:

- синтаксис;
- обмежений функціонал;
- нижча, в порівнянні з аналогами продуктивність;
- можливість пропуску помилок при компіляції.

3.3 Microsoft Azure

Microsoft Azure - це сервіс хмарних обчислень, створена корпорацією Майкрософт для побудови, тестування, розгортання та керування програмами та службами через центри обробки даних, керовані Microsoft. Він надає програмне забезпечення як послугу (SaaS), платформу як послугу (PaaS) та інфраструктуру як послугу (IaaS) та підтримує безліч різних мов програмування, інструментів та рамок,

включаючи як Microsoft, так і програмне забезпечення та системи сторонніх виробників.

Платформа Azure була оголошена в жовтні 2008 року, а загальна комерційна доступність досягла в лютому 2010 року. Спочатку називалася Windows Azure, вона була перейменована в Microsoft Azure в липні 2014 року. З моменту появи сервісу постійно додаються додаткові регіони послуг.

Azure Stack - гібридне хмарне рішення під ключ, пропонуване Microsoft та рядом постачальників обладнання, вперше було оголошено в травні 2015 року. З першим технічним попереднім переглядом у січні 2016 року організації могли використовувати власне обладнання в рамках розгортання Azure Stack. Згодом цей план було згорнуто, і Microsoft вимагала від користувачів придбати попередньо кваліфіковану систему Azure Stack, віривши, що такі пропозиції будуть ефективнішими. Учасники постачальників обладнання постійно випускають нові попередні кваліфіковані системи для використання з Azure Stack.

Під керівництвом генерального директора Microsoft Сатя Наделла Azure розширилася, включаючи підтримку різноманітних дистрибутивів Linux, наявних у віртуальних машинах на платформі Azure. В даний час CentOS, Clear Linux, CoreOS, Debian, Oracle Linux, Red Hat Enterprise Linux, SUSE Linux Enterprise, openSUSE і Ubuntu підтримуються на платформі Azure, а також FreeBSD. Крім того, Azure підтримує зображення Docker.

Microsoft та SAP співпрацюють для того, щоб бізнес-програмне забезпечення та послуги SAP запускалися на Azure. На конференції Sapphire Now у 2018 році Microsoft оголосила про загальну доступність SAP HANA, зазначивши, що Azure пропонує "26 різних пропозицій SAP HANA від 192 ГБ до 24 ТБ". У 2019 році це було продовжено, додавши Amazon Web Services (AWS), Google Cloud та глобальних стратегічних партнерів із обслуговування (GSSP) для доставки SAP HANA на основі хмари.

Azure, як і інші постачальники хмарних послуг, пропонує можливість миттєво надавати обчислювальні ресурси на вимогу. У порівнянні з обтяжливим завданням планування та побудови інформаційного центру на місці, а також необхідних

оновлень обладнання, витрат на обслуговування, вимог до охолодження сервера, витрат на електроенергію та використання площі - особливо для офісів з пов'язаними витратами на нерухомість-- заощадження можуть накопичитися дуже швидко.

Переваги Azure, однак, виходять за рамки контролю витрат. Завдання адміністрування певних технологій, таких як Windows Server, Active Directory та SharePoint, може значно полегшити комбінацію Azure та Office 365. Це звільняє ІТ-співробітників працювати над новими проектами, а не витрачати час на загальне обслуговування системи.

Microsoft наполегливо ухиляється від організацій щодо переміщення AI обчислювальних операцій у Azure. Проект Brainwave - система глибокого навчання на базі FPGA, побудована для AI у режимі реального часу - був випущений як попередній перегляд до Azure на конференції розробників Microsoft Build 2019. Microsoft також додала алгоритми когнітивних служб для отримання розуміння структурованого або неструктурованого контенту, а також додавання функцій до служби машинного навчання Azure, щоб допомогти розробникам використовувати "підхід без коду до створення моделі та розгортання за допомогою нового візуального інтерфейсу машинного навчання". згідно з ZDNet.

3.4 Firebase

Firebase - це платформа для розробки мобільних та веб-додатків, розроблена Firebase Inc. у 2011 році, потім придбана Google у 2014 році. Станом на жовтень 2018 року платформа Firebase налічує 18 продуктів, які використовуються 1,5 мільйона додатків.

Firebase — це BaaS - який почався як запуск стартапу YC11 і перетворився на платформу розробки додатків наступного покоління на хмарній платформі Google.

Дані в реальному часі — це шлях майбутнього. Нічого з цим не порівнюється. Більшість баз даних вимагають здійснення HTTP-дзвінків для отримання та синхронізації даних. Більшість баз даних надають вам дані лише тоді, коли ви просите

їх. Підключаючи додаток до Firebase, ви не підключаєтесь через звичайний HTTP. Ви підключаєтесь через WebSocket. WebSockets набагато, набагато швидше, ніж HTTP. Вам не потрібно здійснювати окремі дзвінки WebSocket, тому що одного з'єднання з сокетом достатньо. Усі ваші дані синхронізуються автоматично через один WebSocket так швидко, як мережа вашого клієнта може переносити його. Firebase надсилає вам нові дані, як тільки вони оновлюються. Коли ваш клієнт зберігає зміну даних, усі підключені клієнти отримують оновлені дані майже миттєво.

Firebase Storage — це потужна, проста та економічно вигідна послуга зберігання об'єктів, створена для масштабів Google. Пакет Firebase SDK для хмарного збереження додає правила безпеки Google для завантаження програм Firebase, незалежно від якості мережі. Можна використовувати SDK для збереження зображень, аудіо, відео чи іншого вмісту, створеного користувачем. На сервері, є можливість використовувати Google Cloud Storage для доступу до тих самих файлів.

Розробники використовують пакети SDK Firebase для Cloud Storage для завантаження та завантаження файлів безпосередньо від клієнтів. Якщо мережевий зв'язок поганий, клієнт може повторити операцію прямо там, де він припинився, економлячи час та пропускну здатність користувачів.

Хмарне збереження зберігає ваші файли для зберігання в хмарі Google, роблячи їх доступними через Firebase та Google Cloud. Це дозволяє гнучко завантажувати файли з мобільних клієнтів через пакети SDK Firebase, а також виконувати обробку на стороні сервера, наприклад фільтрування зображень або перекодувати відео за допомогою Google Cloud Platform. Хмарне збереження масштабується автоматично, це означає, що немає необхідності переходити до будь-якого іншого постачальника.

SDK-файли Firebase для хмарного збереження легко інтегруються з аутентифікацією Firebase для ідентифікації користувачів. Також, можливе використання декларативної мови безпеки, яка дозволяє встановлювати контроль доступу для окремих файлів або груп файлів, щоб могли встановлювати файли як з публічним доступом, так і з приватним.

Більшість додатків повинні знати особу користувача. Знання ідентичності користувача дозволяє додатку надійно зберігати дані користувачів у хмарі та

надавати однаковий персоналізований досвід на всіх пристроях. Перевірка автентичності Firebase надає сервіси, прості у використанні SDK та готові бібліотеки інтерфейсу для автентифікації користувачів в програмі. Він підтримує автентифікацію за допомогою паролів, номерів телефонів, популярних постачальників ідентифікаційних даних, таких як Google, Facebook та Twitter тощо. Автентифікація Firebase тісно інтегрується з іншими службами Firebase і використовує такі галузеві стандарти, як OAuth 2.0 та OpenID Connect, тому вона може бути легко інтегрована у окремий сервер.

Щоб увійти в додаток користувача, спочатку отримуються облікові дані автентифікації від користувача. Ці облікові дані можуть бути електронною адресою користувача та паролем користувача або маркером OAuth від постачальника ідентифікаційних даних. Потім, ці повноваження передаються в SDK для автентифікації Firebase. Потім, сервіси Firebase перевіряють ці дані та повернуть відповідь клієнту. Після успішного входу, відкривається доступ до основної інформації користувача. Також, можна контролювати доступ користувача до даних, що зберігаються в інших продуктах Firebase. Є можливість використовувати наданий маркер автентифікації для перевірки особи користувачів у сервісах сервера. За замовчуванням, автентифіковані користувачі можуть читати та записувати дані в базу даних Firebase Realtime та Google Cloud. Можете контролювати доступ цих користувачів, модифікувавши бази даних та правила безпеки хмарного зберігання.

Cloud function для Firebase дозволяють автоматично запускати резервний код у відповідь на події, викликані функціями Firebase та HTTPS-запитами. Ваш код зберігається в хмарі Google і працює в керованому середовищі. Не потрібно керувати та масштабувати власні сервери.

Після написання та розгортання функцій сервера, Google починають керувати функцією. Є можливість запустити функціонування за допомогою HTTP-запису, або, використовуючи фонові функції, сервіси Google проводять моніторинг програми та використовують цю функцію, коли вона запущена. Коли завантаження зменшується — Google реагує, швидко зменшуючи кількість екземплярів віртуального сервера, необхідних для запуску вашої функції. Кожна функція функціонує у власному

середовищі зі своєю власною конфігурацією.

Життєвий цикл функції:

- Розробник записує код для нової функції, вибираючи постачальника подій (наприклад, база даних в реальному часі) та визначаючи умови, за яких функція повинна виконуватись.
- Розробник розгортає функцію, а Firebase підключає її до обраного постачальника подій.
- Коли постачальник подій генерує подію, яка відповідає умовам функції, викликається код.
- Якщо функція зайнята керуванням багатьма подіями, Google створює більше примірників, щоб швидше обробляти роботу. Якщо функція не працює, екземпляри очищаються.
- Коли розробник оновлює функцію, розгортаючи оновлений код, всі екземпляри старої версії очищаються та замінюються новими екземплярами.
- Коли розробник видаляє функцію, всі екземпляри очищаються, а з'єднання між функцією та провайдером подій видаляється.

Cloud Firestore — це масштабована, хмарна, NoSQL база даних у реальному часі. Для того, щоб структурувати свої дані, визначаються колекції (подібні до таблиць у SQL), які містять документи (подібні до рядків). Кожен документ містить поля, які містять фактичні дані. Є можливість посилатися на окремий документ, використовуючи його унікальний шлях, або можна запитувати колекцію для документів, поля яких містять дані, необхідні для пошуку. Cloud Firestore відрізняється від деяких інших хмарних продуктів, які мають Firebase SDK для мобільних розробок (наприклад, Cloud Functions та Cloud Storage). Продукт Cloud Firestore — це результат співпраці між командами Cloud та Firebase, де поєднуючи досвід та дані в реальному часі, засвоєні ще з баз даних Firebase Realtime, та масштабованість Google Cloud Platform. Cloud Firestore має SDK для мобільних додатків та компонентів сервера, але базовий продукт бази даних той самий.

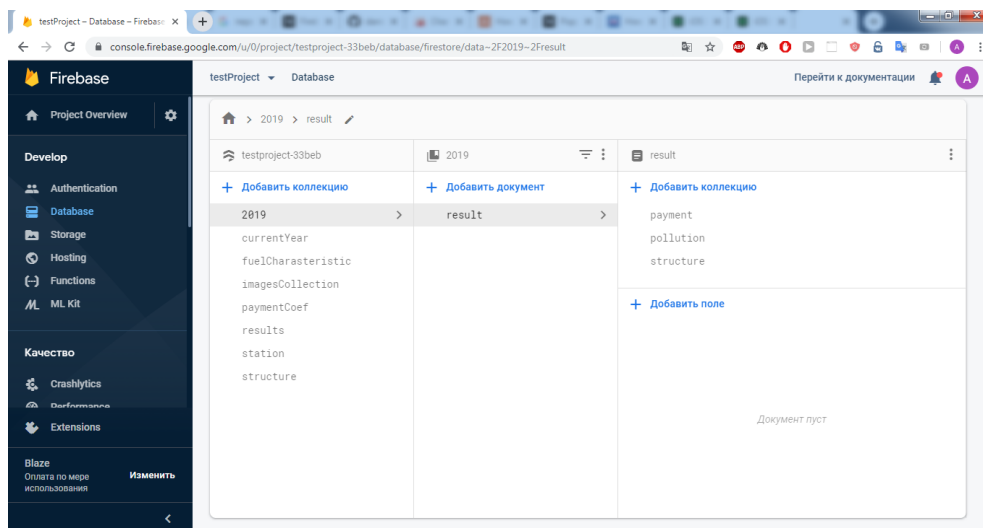


Рисунок 3.2 — Вигляд інтерфейсу та приклад структури Cloud Firestore

Realtime Database є хмарною базою даних типу NoSQL. Дані синхронізуються для всіх клієнтів у режимі реального часу та залишаються доступними, коли програма переходить у режим офлайн. Дані зберігаються в форматі JSON (Рис 4.) і синхронізуються в режимі реального часу з кожним підключеним клієнтом. При створенні крос-платформової програми з пакетами SDK для iOS, Android або JavaScript, всі ваші клієнти діляться одним екземпляром бази даних і автоматично отримують оновлення з найновішими даними.

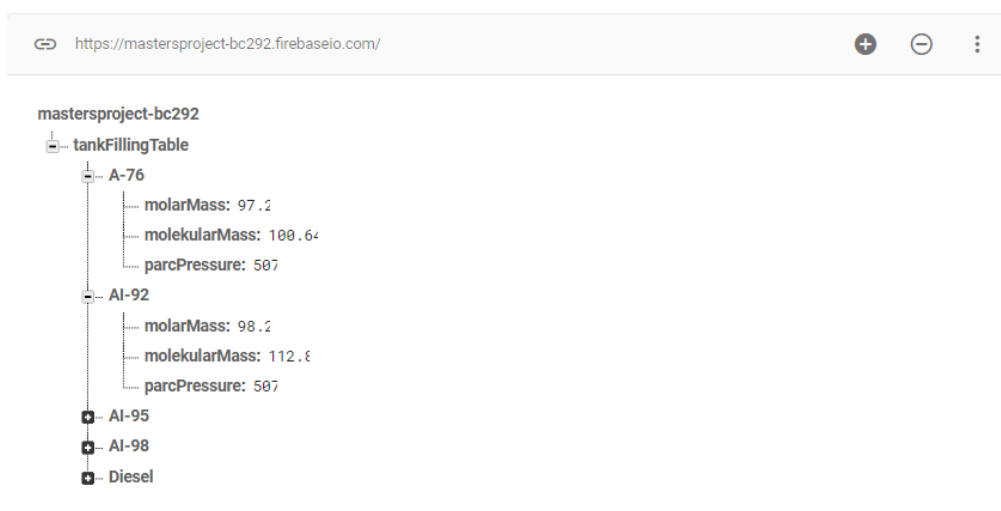


Рисунок 4. – Структура бази даних в Realtime Database

База даних Firebase Realtime дозволяє створювати спільні програми,

дозволяючи безпечний доступ до бази даних безпосередньо з коду клієнта. Дані зберігаються на локальному рівні, і навіть в режимі офлайн події в режимі реального часу продовжують працювати, надаючи кінцевому користувачеві результат. Коли пристрій відновлює з'єднання, база даних синхронізує зміни локальних даних із віддаленими оновленнями, які відбулися під час відключення клієнта, автоматично ліквідуючи будь-які конфлікти. Firebase Realtime забезпечує гнучку мову, засновану на виразах, правила, щоб визначити, як мають бути структуровані дані та коли дані можуть зчитувати або бути збереженими в сховищі. При інтеграції з Firebase Authentication розробники можуть визначити, хто має доступ до даних, і як вони можуть отримати доступ до них. База даних у реальному часі — це база даних NoSQL і як така має різні можливості для оптимізації та функціональні можливості порівняно з реляційною базою даних. Firebase Realtime API розроблений так, щоб дозволяти виконувати операції, які можна швидко виконати. Це дає змогу створити зручну роботу у режимі реального часу, яка може обслуговувати мільйони користувачів без шкоди для роботи та даних. Через це важливо подумати про те, як користувачам отримувати доступ до даних, а потім структурувати їх відповідно до вимог [9].

SwiftCharts — це спеціально кастомізована бібліотека діаграм для iOS, включаючи:

- Bar — звичайний, укладений, згрупований, горизонтальний, вертикальний
- Scatter, Line – пряма, кубічна.
- Area, bubble – наявність кількох осей.
- Candle. Все можна налаштувати, наприклад кольори, види, одиниці, мітки, анімації, взаємодії, осі тощо.

Створення довільних маркерів спрощується за допомогою накладок, перегляду інформації тощо. Використовується модульна архітектура, яка повинна дозволяти легко створювати нові типи діаграм або додавати ефекти.

3.5 Бібліотека Charts

В даний час це стало однією з найпоширеніших рішень більшості аналітичних або будь-яких корпоративних програм, які забезпечують відстеження, перевірку та порівняння даних користувачів. Відображення даних у табличному або колекційному перегляді може бути нудним і дає обмежені можливості візуалізації даних. І з того часу інтеграція діаграм у додатки iOS — це річ, яку слід робити в цих випадках.

В даній бібліотеці можна реалізувати різні типи діаграм:

Bar Charts — коли дані мають різні категорії і потребують порівняння (наприклад, щомісячні цінності продажів), то гістограма забезпечує хорошу та просту візуалізацію.

Pie Charts — коли категоричні дані потрібно відображати у відсотках, хороший вибір будуть кругові діаграми. Кожна діаграма завжди представляє ціле, таким чином, повинна дорівнювати 100%.

Line Charts — візуально представляти зміни за певний проміжок часу, наприклад, інвентар, немає нічого кращого, ніж використання лінійних діаграм. Він показує максимальний і мінімальний бали та показує поточний стан компанії.

Combined-Chart — комбінація лінійної та стовпчикової діаграм.

Scatter Chart — використовує точки для представлення значень для двох різних числових змінних. Положення кожної точки на горизонтальній та вертикальній осі вказує значення для окремої точки даних. Діаграми розсіювання використовуються для спостереження зв'язків між змінними.

CandleStickChart — це стиль фінансової діаграми, що використовується для опису руху цінних паперів, деривативів або валюти. Кожен "свічник", як правило, показує один день, тому одномісячний графік може показувати 20 торгових днів як 20 свічників. Графіки можна також будувати, використовуючи інтервали, менші або довші, ніж один день.

BubbleChart — це тип діаграми, який відображає три виміри даних. Кожна сутність зі своєю триплетою даних: v_1 , v_2 та v_3 . Ці дані пов'язані та побудовані у

вигляді диска, який виражає два значення v через місце диска на координатній сітці x та y , а третє через його розмір.

RadarChart — метод відображення багатоваріантних даних у вигляді двовимірної діаграми з трьох або більше кількісних змінних, представлених на осях, починаючи з тієї ж точки.

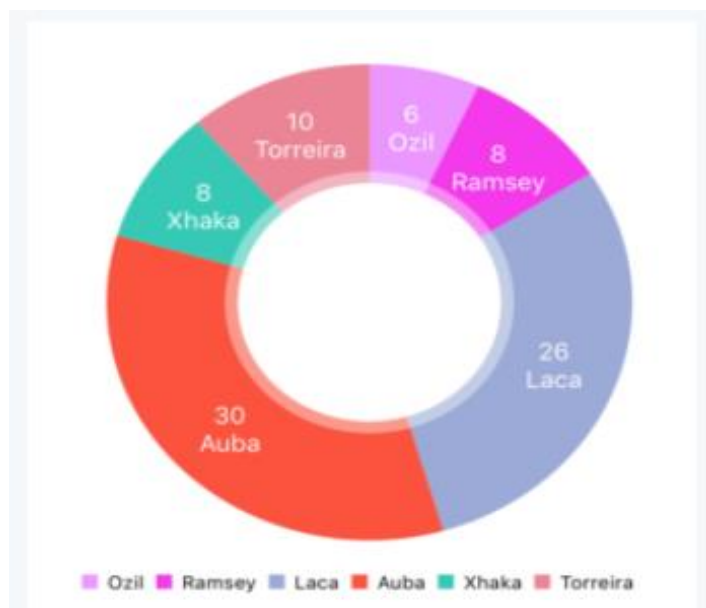


Рисунок 3.3 — Приклад діаграми

Отже, бібліотека пропонує наступні можливості:

- 8 різних типів діаграм;
- масштабування на обох осях;
- експорт діаграми в форматі PNG або JPEG;
- заздалегідь визначені кольорові шаблони;
- граничні лінії для надання додаткової інформації;
- самостійне налаштування за наступних параметрів: відтінки, шрифти, легенди, кольори, фон, жести, пунктирні лінії;
- завантаження даних безпосередньо з мобільної бази даних Realm.io.

3.5 Середовище емуляції

Запуск та тестування мобільного додатку було здійснене на емуляторі Appetize.io.

Я включив Appetize.io і перерахував його в першу чергу завдяки простоті користування, яку пропонує ця послуга. Враховуючи дефіцит емуляторів iOS для ПК та Mac, в першу чергу, найпростішим є рішення, яке потребує завантаження додатку на веб-сайт та його використання без додаткових налаштувань.

Розробники можуть використовувати Appetize.io для запуску емульованих iOS додатків у браузері з метою вбудовування додатків на веб-сайти, тестування чи його використання на інших фазах розробки. Якщо говорити про ціни, то безкоштовний період цього симулятора iPhone хоч і обмежений, але задовольняє умовам використання на незначних часових проміжках і щомісячно оновляється. Можна відстежувати своє використання за допомогою інформаційної панелі та налаштовувати сповіщення, коли ви досягнете межі.

Переваги даного середовища:

- запуск додатків iOS та Android в браузері на будь-якому комп'ютері;
- можливість вибору будь-якого девайсу, операційної системи та версії додатку;
- простота в використанні та підтримці;
- відстеження та аудит використання агентом, точне налаштування контролю доступу та дозволів.

Недоліки:

- обмежена кількість часу безоплатного використання протягом місяця.

Electric Mobile Studio — це платний спосіб тестування та запуску програм iOS на комп'ютерах Windows, проте вона пропонує повну пробну версію продукту протягом 7 днів, що робить неможливим її безоплатне використання на довготривалому проміжку. Його основні особливості включають повноцінну емуляцію iPhone, iPad та можливість використовувати один і той же продукт на двох

машинах для загальних робочих налаштувань в офісі та будинку.

Розробники також віддають перевагу цьому емулятору iPhone для Windows 10 завдяки інтегрованим інструментам налагодження WebKit та Chrome, що полегшують їх роботу. Можна додати клавіші навігаційного клавіші для відображення улюблених ярликів. Користувачі Windows також можуть легко інтегрувати цей інструмент для додатків iOS із Visual Studio.

Висновки до розділу 3

В якості середовища розробки було обрано Xcode. Це середовище підтримується компанією Apple, є основним та рекомендованим для розробки нативних додатків для платформ iOS та macOS. Враховуючи можливості та популярність середовища, альтернативні варіанти не є вагомими конкурентами.

Між мовами Swift та Objective-C є багато спільного, проте у мови Swift є багато переваг в порівнянні не тільки з мовою-попередником, але й з іншими мовами. Swift вже давно є популярнішою в порівнянні з Objective-C. Більшість нових проєктів та рішень реалізується саме на Swift. Станом на 2019 рік, мова Swift є найоптимальнішим варіантом для розробки нативних додатків на платформу iOS

Серед хмарних платформ для розробки програмного забезпечення було обрано Firebase, оскільки вона має ряд переваг та є рекомендованим інструментом для розробки мобільних додатків. Firebase надає можливість обрати поміж двох хмарних сховищ даних, серед яких Cloud Firestore та Realtime Database. Було обрано Cloud Firestore, оскільки дане сховище є вдосконаленим в порівнянні з Realtime Database сховищем та рекомендоване для нових проєктів.

Для графічного представлення даних, було обрано бібліотеку Charts. Вона є безоплатною та пропонує вичерпний функціонал, необхідний для різнопланового представлення даних.

Для запуску та тестування додатку була використана середовище Appetize.io. Воно є потужним та доступним рішенням, що може запускатись в браузері.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Програма була розроблена на мові Swift з використанням хмарної СУБД Firebase та бібліотки Charts. В розділі будуть вказані реалізовані програмні модулі, підключені бібліотеки, проектування сховища даних, описаний архітектурний шаблон проекту та вказані результати тестування проекту.

4.1 Програмна реалізація

На рисунку 4.1 представлені наступні програмні модулі:

- модуль авторизації/реєстрації
- модуль обчислення викидів при заповненні цистерни
- модуль обчислення викидів при заправці автомобіля;
- модуль обчислення викидів при збереженні палива;
- модуль заповнення податкової декларації;
- модуль перегляду складу мережі АЗС;
- модуль перегляду аналітичних даних;
- модуль доповнення складу АЗС
- модуль видалення станцій АЗС
- модуль оновлення мережі АЗС;
- модуль супроводу бази даних;

На рисунку 4.1 зображені модулі мережі АЗС:

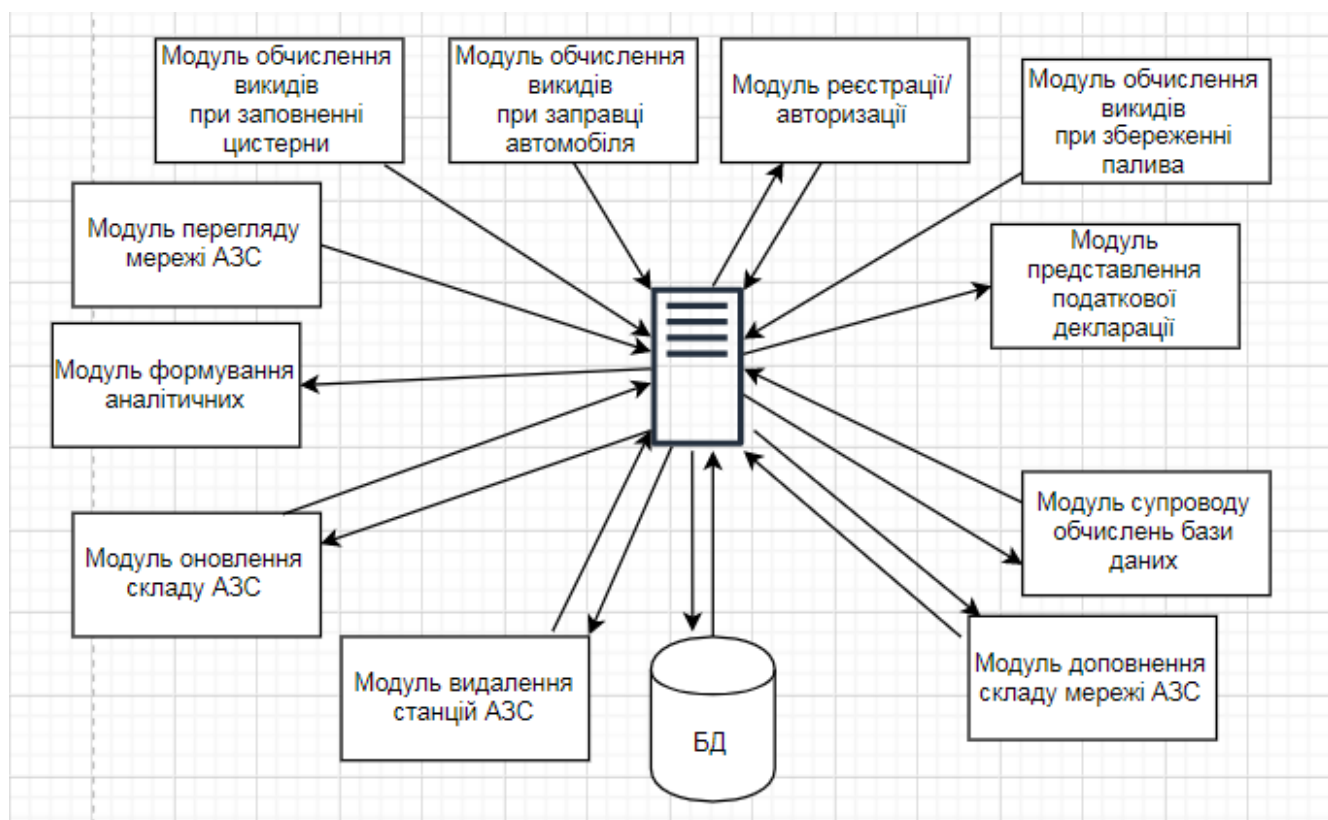


Рисунок 4.1 – Основні модулі програми

На рисунку 4.2 зображена діаграма прецедентів додатку.

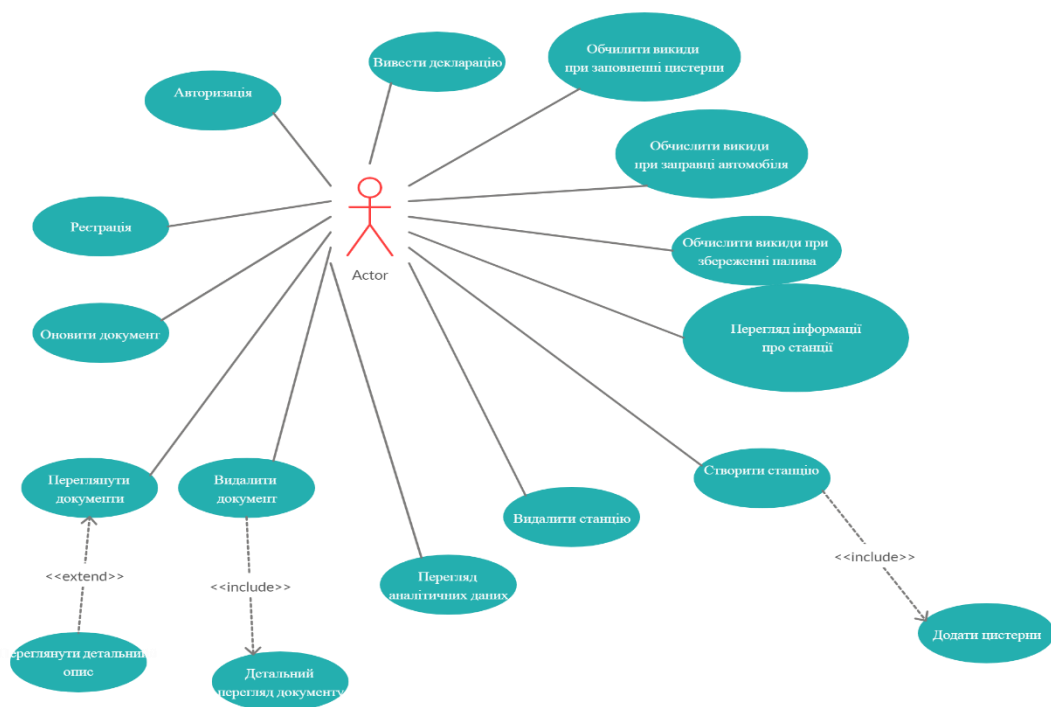


Рисунок 4.2 – Діаграма прецедентів програми

На рисунку 4.3 зображена блок-схема обчислень викидів

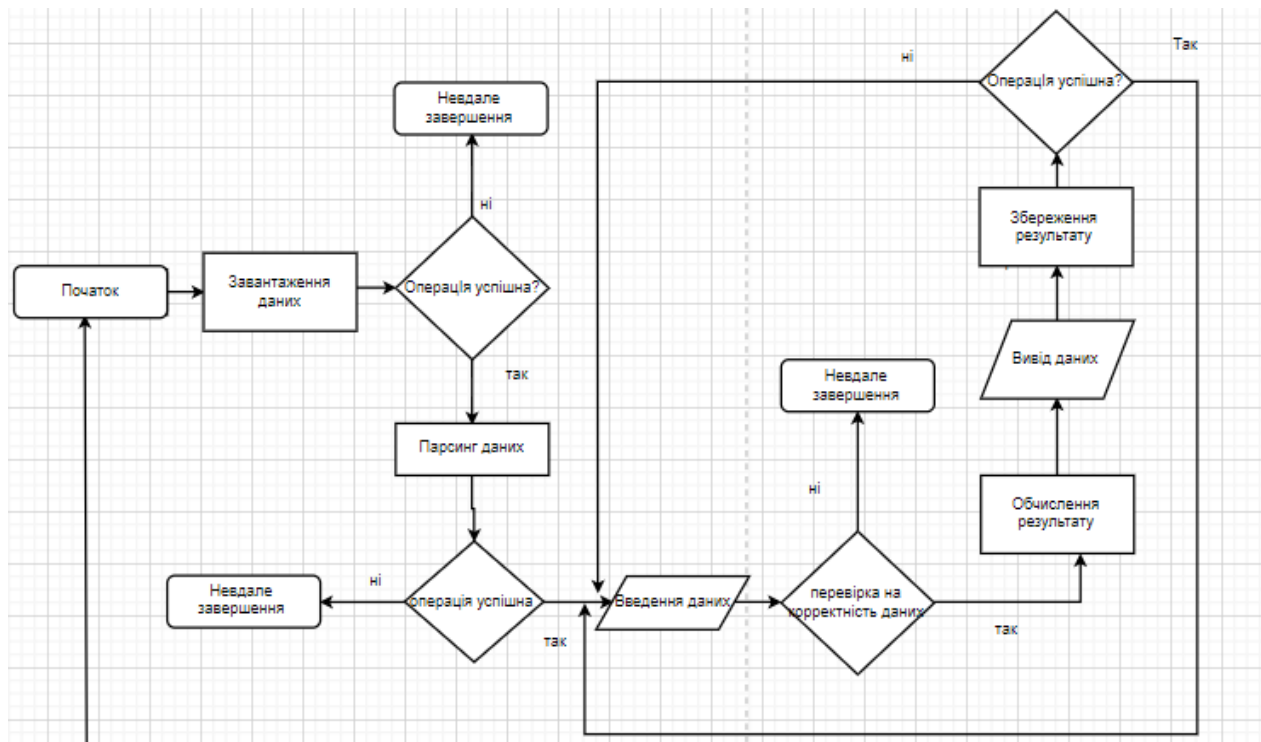


Рисунок 4.3 — Обчислення викидів

На рисунку 4.4 зображена блок-схема актуалізаціх складу мережі АЗС

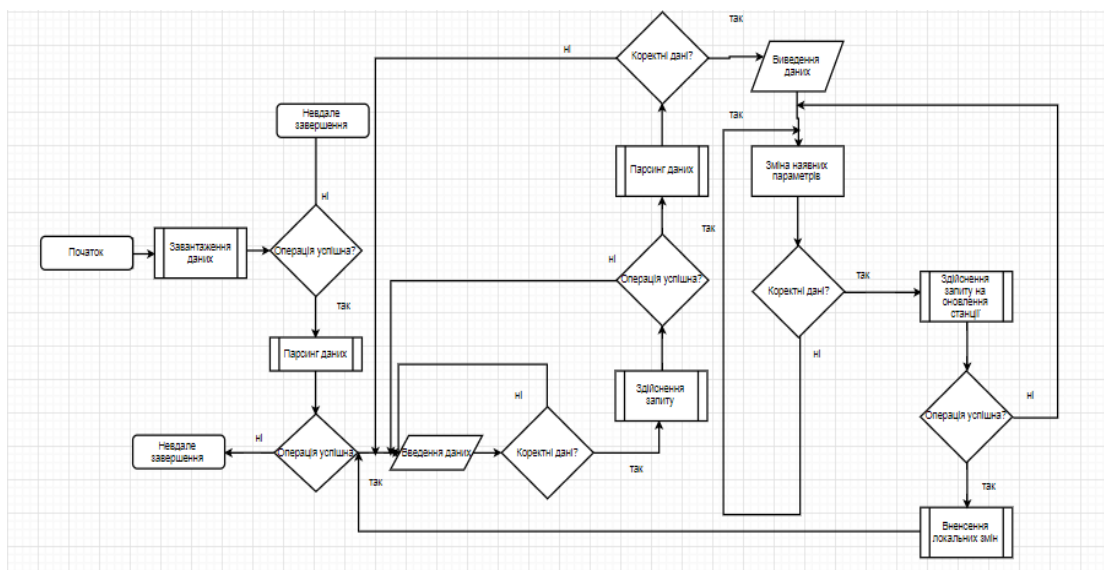


Рисунок 4.4 — Оновлення станції

На рисунку 4.5 зображена блок-схема видалення станції мережі АЗС

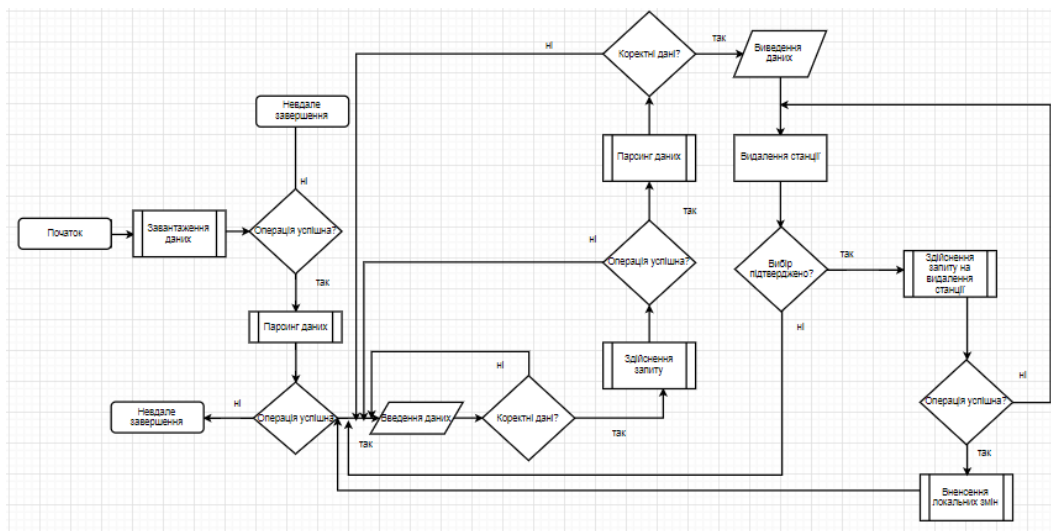


Рисунок 4.5 — Видалення станції мережі

На рисунку 4.6 зображена блок-схема створення станції мережі АЗС

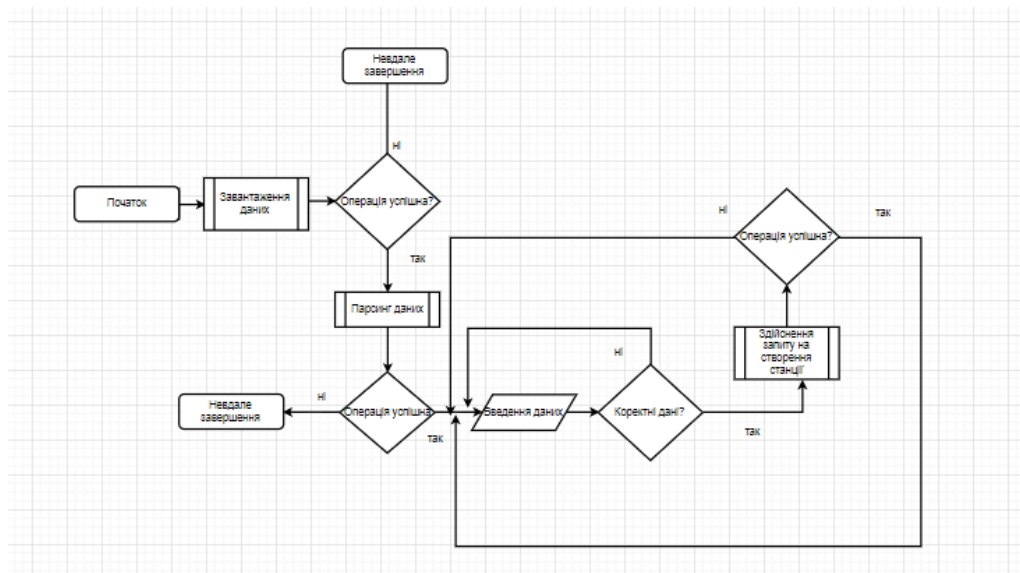


Рисунок 4.6 — Створення нової станції мережі

На рисунку 4.7 зображена блок-схема видалення станції мережі АЗС

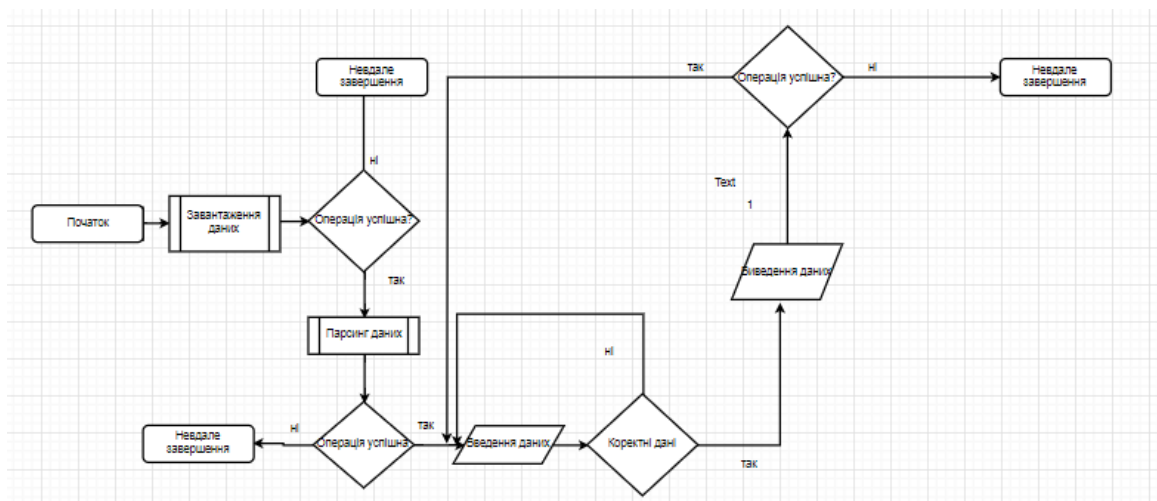


Рисунок 4.7 — Виведення даних станції

На рисунку 4.8 зображена блок-схема видалення документу АЗС

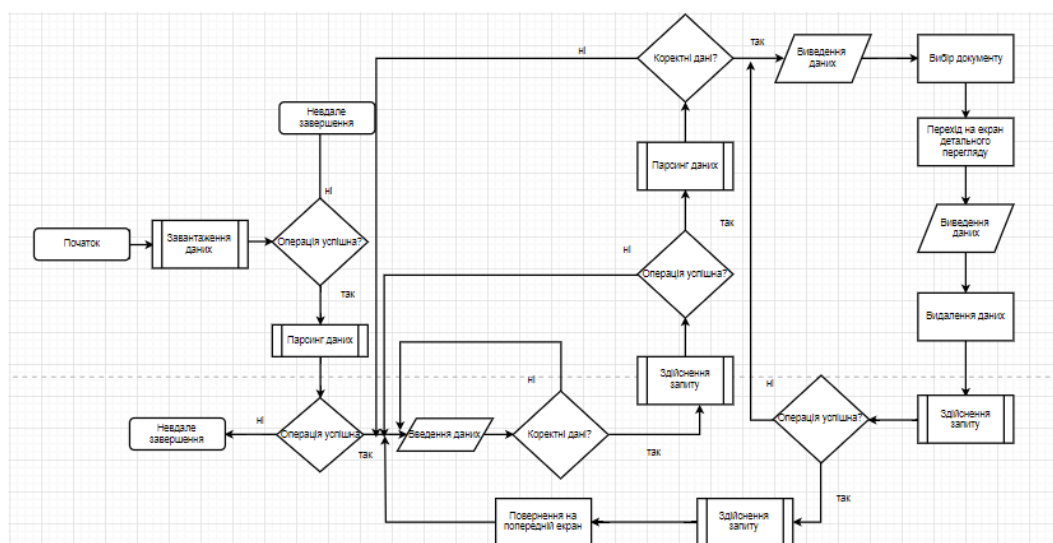


Рисунок 4.8 — Видалення документу

На рисунку 4.9 зображена блок-схема перегляду запису АЗС

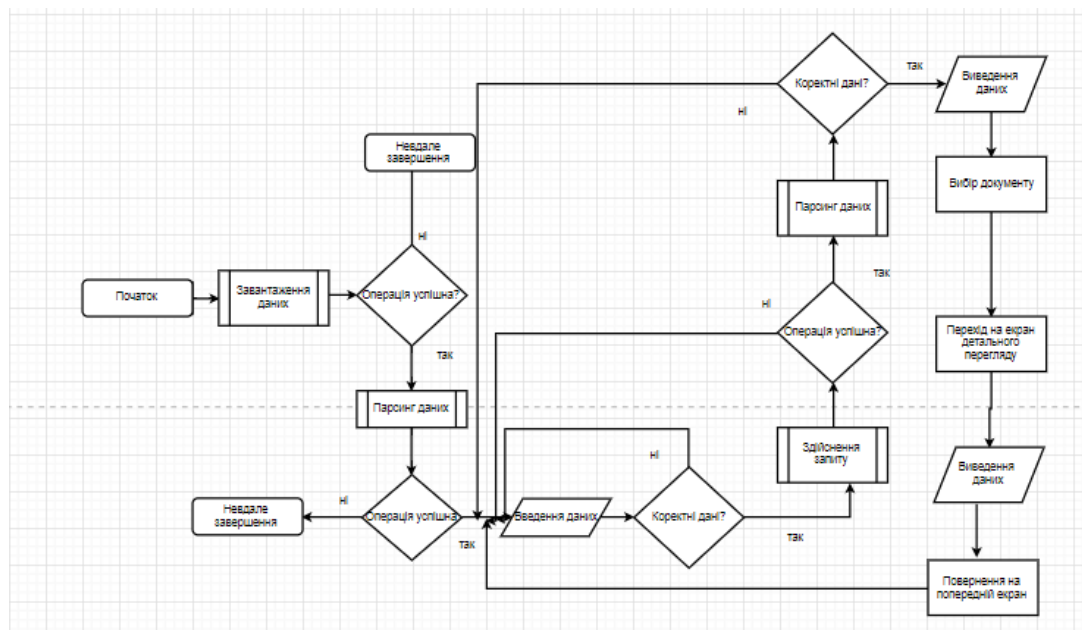


Рисунок 4.9 — Перегляд документу

На рисунку 4.10 зображена блок-схема перегляду запису АЗС.

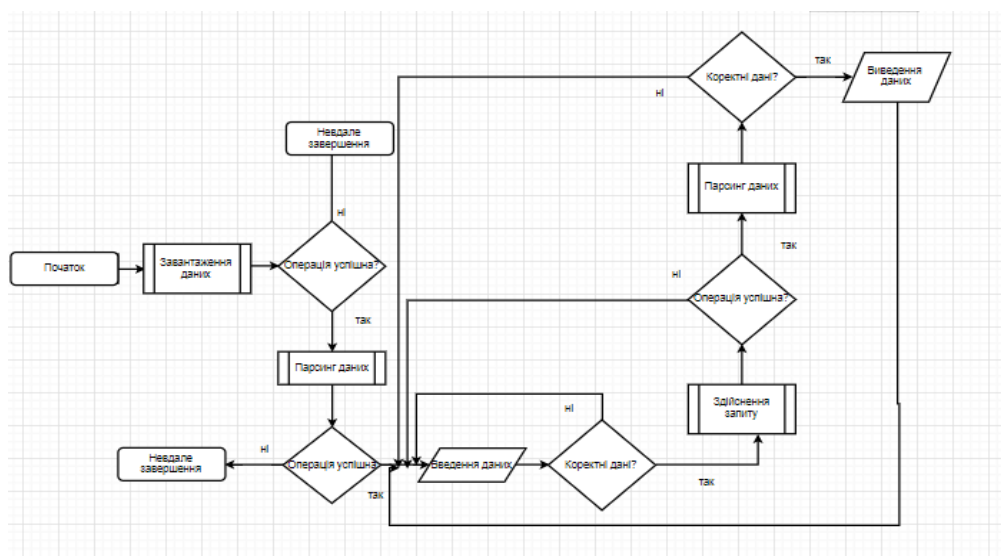


Рисунок 4.10 — Перегляд даних станції

На рисунку 4.11 зображена блок-схема перегляду аналітичних даних

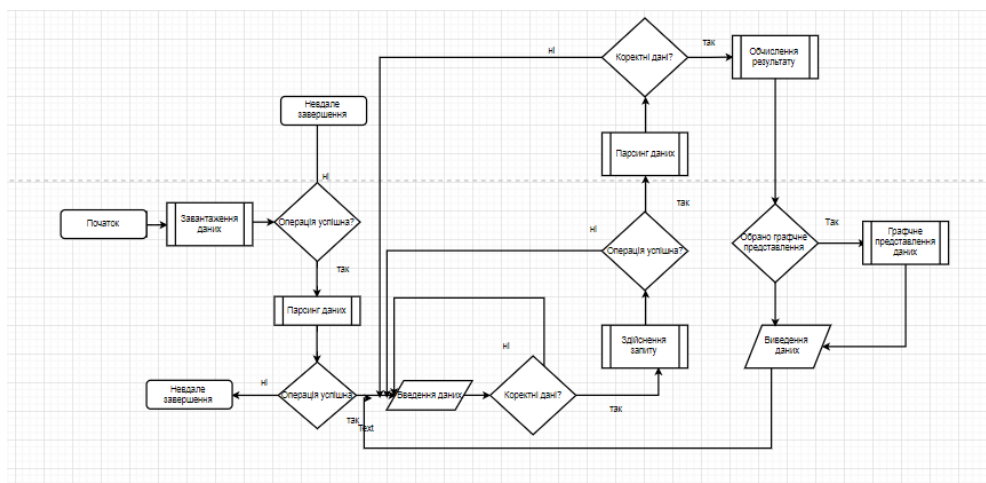


Рисунок 4.11 — Перегляд аналітичних даних

Далі будуть вказані бібліотеки, що були підключені для розширення можливостей програми (рисунок 4.12).

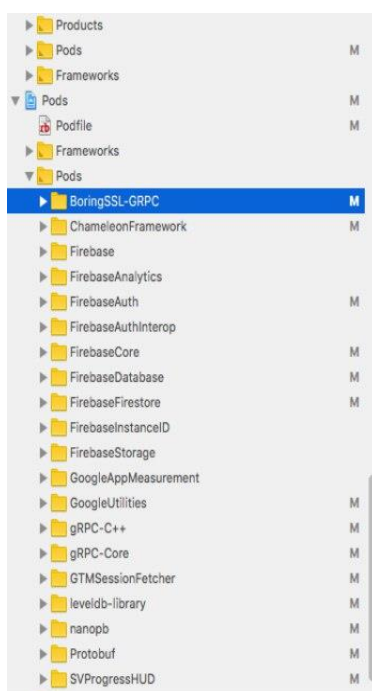


Рисунок 4.12 — Підключені бібліотеки проекту

BoringSSL-GRPC – це виделка OpenSSL, розроблена для задоволення потреб Google.

ChameleonFramework – легкий фреймворк для iOS, що працює з мовами Objective-C та Swift та дозволяє полегшити роботу з налаштуванням кольорів. Він

побудований на ідеї, що програмні програми повинні функціонувати без значних ресурсних затрат, одночасно підтримуючи свої інтерфейси

Firestore – SDK для роботи платформи для розробки мобільних додатків на платформі iOS.

FirestoreAnalytics – демонструє збір аналітики щодо роботи додатку через API Analytics.

FirestoreAuth – перевірка автентичності Firestore надає сервіси, прості у використанні SDK та готові бібліотеки інтерфейсу для автентифікації користувачів програми. Він підтримує автентифікацію, використовуючи паролі, номери телефонів, популярних постачальників ідентифікаційних даних, таких як Google, Facebook та Twitter тощо.

FirestoreCore – займає ключове місце у зв'язку між різними продуктами Firestore, установленними в додатку.

FirestoreDatabase – зберігає та синхронізує дані з хмарною базою даних NoSQL. Дані синхронізуються для всіх клієнтів у режимі реального часу та залишаються доступними, коли програма переходить у режим офлайн

FirestoreFirestore – гнучка, масштабована хмарна база даних NoSQL нового покоління, що дозволяє зберігати та синхронізувати дані для розробки клієнтських та серверних служб.

FirestoreInstanceID – надає унікальний ідентифікатор для кожного екземпляра програми та механізм автентифікації та авторизації дій, наприклад: надсилання повідомлень FCM.

FirestoreAppMeasuremenet – містить класи, які налаштовують основні сервіси Firestore Analytics.

GoogleAppUtilities – Спільні методи мережевої роботи для бібліотек Google. Призначений не для прямого використання, а як залежність деяких бібліотек Google.

gRPC++ — фреймворк, що надає можливість викликати методи серверного додатку іншої машині як локальний об'єкт, що полегшує створення розподілених додатків та служб. Як і у багатьох системах RPC, gRPC++ базується на ідеї визначення послуги із зазначенням методів, які можна викликати віддалено з їх параметрами та

типами повернення. На стороні сервера сервер реалізує цей інтерфейс і запускає сервер gRPC++ для обробки дзвінків клієнтів. На стороні клієнта присутня заглушка, хоча в деяких мовах її називають просто клієнтом, яка забезпечує ті самі методи, що і сервер

GMTCSessionFetcher – спрощує виконання http-операцій для додатків Cocoa. Застосунок реалізований як обгортка для NSURLSession, тому його поведінка є асинхронною і використовує параметри операційної системи на iOS та Mac OS .

Leveldb-library – це швидка бібліотека зберігання типу ключ-значення. Розроблена компанією Google, вона забезпечує впорядковане відображення від рядкових ключів до значень рядка.

Nanopb – це бібліотека ANSI-C для кодування та декодування повідомлень у форматі буферів протоколів Google з мінімальними вимогами до оперативної пам'яті та кодового простору. Він в першу чергу підходить для 32-бітних мікроконтролерів.

Protobuf – це платформи-нейтральний, розширюваний механізм, розроблений компанією Google для серіалізації структурованих даних — подібний до XML, але менший, швидший та простіший. Користувач визначає, як структурувати свої дані один раз, тоді можна використовувати спеціальний згенерований вихідний код, щоб легко писати та читати структуровані дані з різних потоків даних та з використанням різних мов.

SVProgressHUD – це чистий і простий у використанні HUD, призначений для відображення ходу поточного завдання на iOS та tvOS.

4.2 Збереження даних

На відміну від реляційних баз даних, в Cloud Firestore немає ні таблиць, ні рядків. Оскільки це декументо-орієнтована база даних, дані зберігаються в документах(рис 4.13), які організовані в колекції.

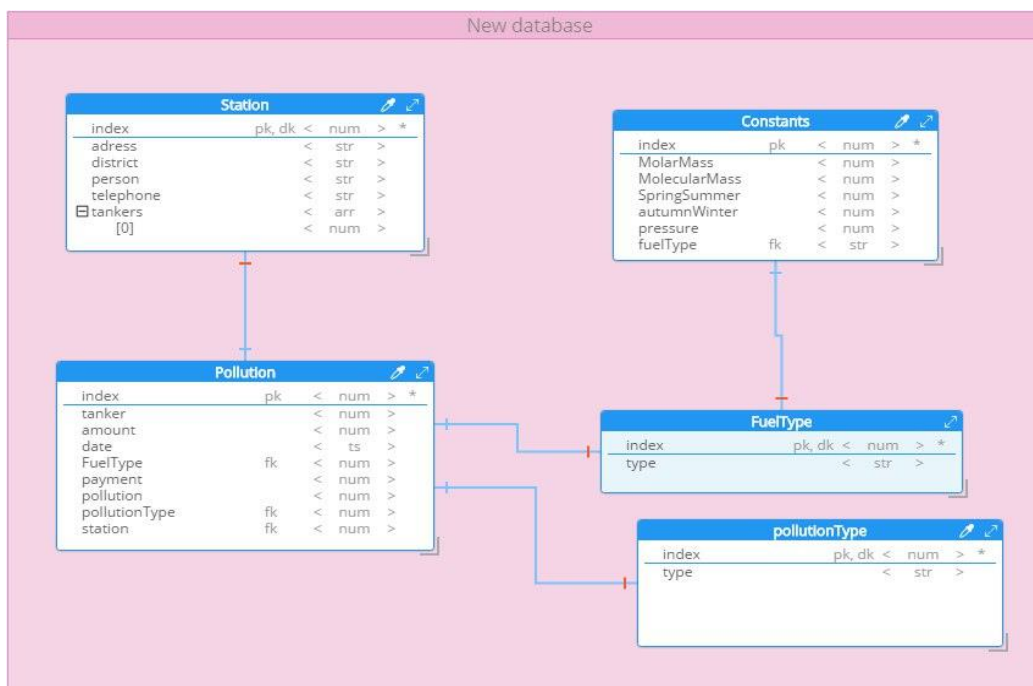


Рисунок 4.13 — Схема сховища даних

Розглянемо склад бази даних проекту:

Документи колекції Pollution, що містять дані про викиди з АЗС, складаються з наступних полів:

- index — індекс документу;
- tanker — цистерна, в якій зберігалось паливо;
- amount — об'єм палива;
- date — календарний місяць;
- fuelType — тип палива;
- payment — плата за викид;
- pollution — об'єм забруднення;
- pollutionType — тип забруднення.

Документи колекції Stations, що містять загальні дані щодо АЗС, складаються з наступних полів:

- index — індекс документу;
- address — адреса;
- tankers — цистерни;
- district — район;

- person — контактна особа;
- station — індекс станції;
- telephone – телефон.

Документи колекції FuelEnzu містять наступні поля;

- index — індекс документу;
- type – тип палива.

Документи колекції Constants містять наступні поля:

- index — індекс документу;
 - fuelType – тип палива;
 - molecularMass – молекулярна маса;
 - molarMass – молярна маса;
 - springSummer – константа в весняно-літній період;
 - autumnWinter – температура в осінньо-зимовий період.
 - Pressure – тиск рідини
- Документи колекції pollutionType містять наступні поля:
- index — індекс документу;
 - type – тип палива;

4.3 Архітектурний шаблон

Архітектурний шаблон Model-View-Controller (MVC) призначає об'єктам у програмі одну з трьох ролей: Model, View або Controller. Шаблон визначає не тільки рольові об'єкти, які є у програмі, він визначає тип зв'язку об'єктів між собою. Кожен з трьох типів об'єктів відокремлений від інших абстрактними межами і зв'язаний з об'єктами інших типів через ці межі. Колекцію об'єктів певного типу MVC у додатку іноді називають шаром — наприклад, шаром моделі. MVC є основним архітектурним шаблоном при роботі зі фреймворком Cocoa. Існує багато переваг використання цього шаблону. Програми, що мають дизайн MVC, простіше розширюються[8]. Більше того, багато технологій та архітектур фреймворка Cacao базуються на MVC.

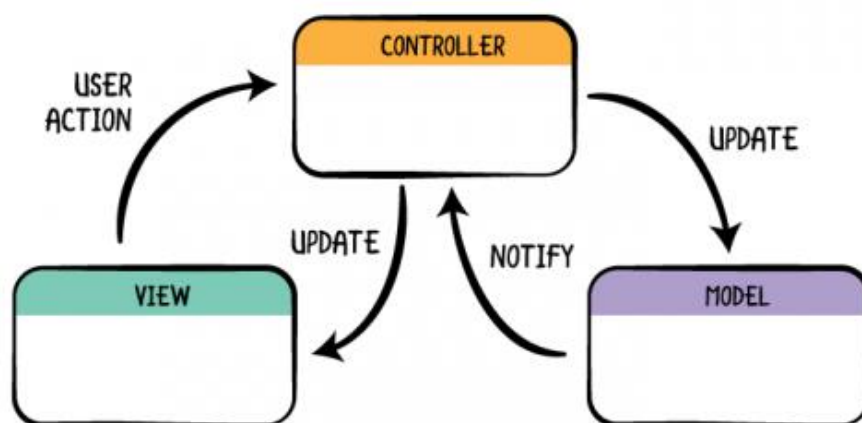


Рисунок 4.14 – зображення моделі MVC

Об'єкти ролі Model інкапсулюють дані, специфічні для програми та визначають логіку та обчислення, які маніпулюють та обробляють ці дані. Наприклад, модельний об'єкт може представляти персонажа в грі або контакту в адресній книзі. Значна частина даних, що є частиною постійного стану програми, повинна знаходитися в модельних об'єктах після завантаження даних у додаток. Оскільки об'єкти моделі представляють знання та досвід, пов'язаний із конкретною проблемною областю, вони можуть бути використані повторно в подібних проблемних областях. В ідеалі, об'єкт моделі не повинен мати явного зв'язку з об'єктами перегляду, які представляють його дані та дозволяють користувачам редагувати ці дані — він не повинен стосуватися питань користувацького інтерфейсу та презентації.

Дії користувача на рівні перегляду, які створюють або змінюють дані, передаються через об'єкт типу Controller і призводять до створення або оновлення Model об'єкта. Коли об'єкт Model змінюється, він повідомляє об'єкт Controller, який оновлює відповідні об'єкти перегляду.

Об'єкт типу View — це об'єкт у програмі, який користувачі можуть бачити. Об'єкт перегляду вміє малювати себе і може реагувати на дії користувача. Основна мета об'єктів перегляду — відображення даних. Незважаючи на це, об'єкти перегляду зазвичай знімаються з модельних об'єктів у програмі MVC.

Перегляд об'єктів дізнається про зміни в модельних даних через об'єкти

контролера програми та передає ініційовані користувачем зміни (наприклад, текст, введений у текстове поле — через об'єкти контролера до об'єктів моделі програми).

Controller є зв'язуючою ланкою між різними об'єктами Model та View. Об'єкти ролі Controller отримавши дані про зміни в Model, повідомляють про це View, та навпаки. Серед інших функцій, можна вказати наступні пункти: управління життєвим циклом об'єктів, виконувати установку та координацію завдань для програми. Controller інтерпретує дії користувача, зроблені в об'єктах перегляду, і передає нові або змінені дані на рівень моделі. Коли об'єкти моделі змінюються, об'єкт контролера передає ці нові дані моделі об'єктам подання, щоб вони могли їх відображати.

4.4 Тестування

В таблиці 4.1. було здійснено тестування процесу аутентифікації користувача.

Таблиця 4.1. Аутентифікація

Мета тесту	Перевірка можливості авторизації користувача в системі
Початковий стан	Вхід в програму
Вхідні дані	Логін та пароль
Схема проведення тесту	Введення логіна та пароля користувача
Очікуваний результат	Успішний вхід в систему
Стан програмного продукту після проведення випробувань	Перехід на головний екран додатку

В таблиці 4.2. був здійснений тестування процесу реєстрації користувача.

Таблиця 4.2. Реєстрація користувача

Мета тесту	Перевірка можливості реєстрації користувача в системі
Початковий стан	Екран з незаповненими текстовими полями
Вхідні дані	Логін, пароль та підтвердження паролю
Схема проведення тесту	Введення даних та натиск кнопки реєстрації
Очікуваний результат	Повідомлення про успішну реєстрацію користувача
Стан програмного продукту після проведення випробувань	Перехід на екран авторизації

В таблиці 4.3 було здійснено тестування процесу обчислення викидів.

Таблиця 4.3. Обчислення викидів

Мета тесту	Перевірка результату обчислень викидів
Початковий стан	Вікно обчислення викидів при заповненні палива
Вхідні дані	Параметри обчислень: індекс станції, часовий проміжок, індекс цистерни, об'єм палива
Схема проведення тесту	Введення параметрів та обчислення результату
Очікуваний результат	Об'єм забруднення
Стан програмного продукту після проведення випробувань	Вивід результату на екран та збереження в базі даних.

В таблиці 4.4. було здійснено тестування процесу перегляду даних станції.

Таблиця 4.4. Перегляд даних станції

Мета тесту	Перевірка результату запиту щодо даних станції
Початковий стан	Екран перегляду даних станції
Вхідні дані	Індекс станції
Схема проведення тесту	Вибір індексу та перегляд даних про станцію
Очікуваний результат	Виведення даних
Стан програмного продукту після проведення випробувань	Вивід результату на екран в вигляді таблиці.

В таблиці 4.5 було здійснений тестування процесу запитів до бази даних.

Таблиця 4.5. Перегляд записів бази даних

Мета тесту	Перевірка коректності виводу інформації
Початковий стан	Екран перегляду даних про викиди та виплати
Вхідні дані	Індекс станції, часовий проміжок
Схема проведення тесту	Вибір індексу станції та часового проміжку. Перехід на екран перегляду детальної інформації
Очікуваний результат	Виведені результати запитів та можливість детального опису
Стан програмного продукту після	

В таблиці 4.7 було здійснено тестування процесу перегляду даних для заповнення податкової декларації.

Таблиця 4.7. Перегляд даних для заповнення податкової декларації

Мета тесту	Перевірка коректності виводу інформації
Початковий стан	Екран перегляду даних про викиди та виплати
Вхідні дані	Індекс станції, часовий проміжок
Схема проведення тесту	Вибір індексу станції та часового проміжку. Перехід на екран перегляду детальної інформації
Очікуваний результат	Виведені результати запитів та можливість детального опису
Стан програмного продукту після проведення випробувань	Виведені результати за вказаним параметрами

В таблиці 4.8 було здійснено тестування процес перегляду аналітичних даних.

Таблиця 4.8. Перегляд аналітичних даних

Мета тесту	Перевірка аналітичних даних
Початковий стан	Екран перегляду аналітичних даних
Вхідні дані	Індекс станції, часовий проміжок, тип аналітичних даних
Схема проведення тесту	Вибір індексу станції, часового проміжку та тип аналітичної інформації. Перегляд інформації в графічному та текстовому вигляді

Продовження таблиці 4.8

Очікуваний результат	Вивід інформації та можливість перегляду інформації в різному вигляді
Стан програмного продукту після проведення випробувань	Вказана інформація виведена на екран

В таблиці 4.9 було здійснено тестування процесу створення станції.

Таблиця 4.9 Створення станції

Мета тесту	Створення нової станції
Початковий стан	Екран створення нової станції
Вхідні дані	Адреса, район, телефон, персону, індекс та тип палива
Схема проведення тесту	Введення параметрів нової станції, перехід на екран для введення нових цистерн
Очікуваний результат	Повідомлення про успішне створення нової станції
Стан програмного продукту після проведення випробувань	Вивід результату на екран та збереження в базі даних.

В таблиці 4.10 було здійснено тестування процесу видалення станції.

Перейшовши на екран видалення станції, користувач зможе ввести індекс станції та переглянути детальну інформацію щодо неї. Користувач матиме можливість видалити дану станцію зі сховища даних. Після її видалення в полі індексів буде відсутній індекс видаленої станції

Таблиця 4.10. Видалення станції

Мета тесту	Перевірка результату видалення станції
------------	--

Початковий стан	Здійснено перехід на екран видалення станцій
Вхідні дані	Індекс станції
Схема проведення тесту	Введення параметру та виконання видалення, перегляд інформацій та позитивна відповідь на повідомлення про видалення станцій
Очікуваний результат	Повідомлення про видалення
Стан програмного продукту після проведення випробувань	Повідомлення про видалення станції

В таблиці 4.11 було здійснено тестування процесу видалення документів.

Таблиця 4.11. Видалення документів сховища даних

Мета тесту	Видалення вказаної інформації з бази даних
Початковий стан	Вікно обчислення викидів при заповненні палива
Вхідні дані	Параметри обчислень: індекс станції, часовий проміжок, тип забруднення
Схема проведення тесту	Введення параметрів, видалення вказаних даних
Очікуваний результат	Відсутність раніше видалених даних при виведенні результату обчислень
Стан програмного продукту після проведення випробувань	Вивід результату на екран та збереження в базі даних.

В таблиці 4.12 було здійснено тестування процесу оновлення даних станції.

Таблиця 4.12. Оновлення даних станції

Мета тесту	Оновлення даних станцію
Початковий стан	Вікно обчислення викидів при заповненні палива
Вхідні дані	Параметри обчислень: індекс станції, часовий проміжок, індекс цистерни, об'єм палива
Схема проведення тесту	Введення параметрів та обчислення результату
Очікуваний результат	Об'єм забруднення
Стан програмного продукту після проведення випробувань	Вивід результату на екран та збереження в базі даних.

Висновки до розділу 4

В розділі були описані основні деталі програмної реалізації, результат проектування сховища даних, описаний архітектурний паттерн та проведене тестування програмного продукту. Провівши тестування всіх екранів, програмний продукт був мануально протестований на коректність роботи.

5 РОБОТА КОРИСТУВАЧА З ПРОГРАМОЮ

В розділі буде розглянута робота користувача з програмою. Після опису інсталяції програми, буде вказаний опис роботи користувача з програмним продуктом, що супроводжуватиметься зображеннями екрану при вказаному моменті роботи програми.

5.1 Інсталяція програми

Мінімальні системні вимоги:

- Операційна система — iOS 9.0
- Оперативна пам'ять – 1 ГБ

Можливе завантаження з AltStore.

5.2 Вхід в систему

При запуску додатку, перед користувачем постає екран авторизації з текстовими полями для введення логіну та пароля, а також кнопка для входу в систему. В правому верхньому куті екрану присуіная кнопка реєстрації, з допомогою якої, користувач може перейти на екран реєстрації зареєструватись в системі.

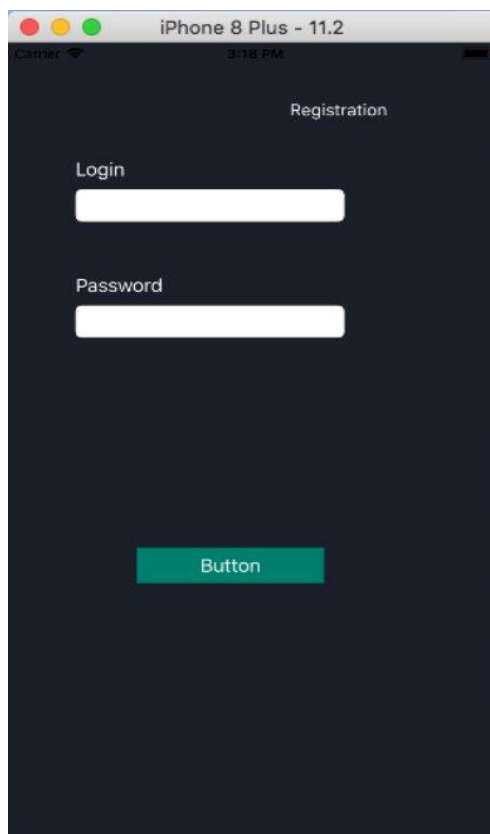


Рисунок 5.1 — Авторизація користувача

В вікні реєстрації користувача присутні кнопка реєстрації та текстові поля для введення логіну, пароля та підтвердження пароля. Вимоги до введення даних - логін повинен бути унікальним та не повторюватись серед наявних збережених логінів, тоді як пароль повинен містити як мінімум 4 знаки. Після завершення введення даних та у випадку коректності введених даних, користувач буде зареєстрований та зможе авторизуватись в додатку за вказаними при реєстрації дані.

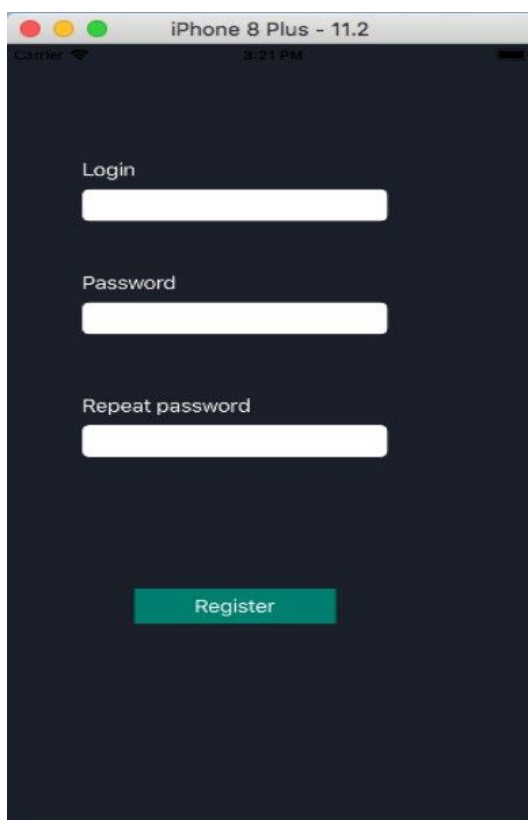


Рисунок 5.2 — Реєстрація користувача

5.3 Обчислення викидів

Для обчислення викидів, в програмі присутні 3 вікна: викиди при заповненні резервуару, викиди при заправці автомобіля і викиди при збереженні палива.

Користувачу необхідно ввести індекс станції, місяць за який відбувався викид, індекс резервуару та кількість пального за даний місяць. Результатом обчислення буде викид пального за даний період часу. Аналогічними є вікна обчислення для обчислення викидів при розповсюдженні пального та його збереженні

The screenshot shows a mobile application interface titled "Tank filling". It features several input fields: "Station" with the value "1", "Date" with "10/2019", "Tanker" with "2", and "Amount" with "30,0". Below these fields is a green button labeled "Calculate". Under the button is a "Result" field displaying "0.073". At the bottom of the screen is a navigation bar with three tabs: "Calculations" (highlighted in blue), "Data", and "Manager". The status bar at the top indicates "Carrier", signal strength, and the time "8:29 PM".

Рисунок 5.3 — Введення вхідних даних про тип викиду

5.3 Робота з даними

При роботі з даними є 4 опції: вивід інформації про станції мережі АЗС, вивід даних про викиди та виплати, заповнення декларації та представлення даних в графічному вигляді.

У вкладці Stations є можливість переглянути інформацію про станцію мережі АЗС. Там є інформація про район, де розташована АЗС, контактна особа, її адреса, та склад цистерн на даній АЗС.

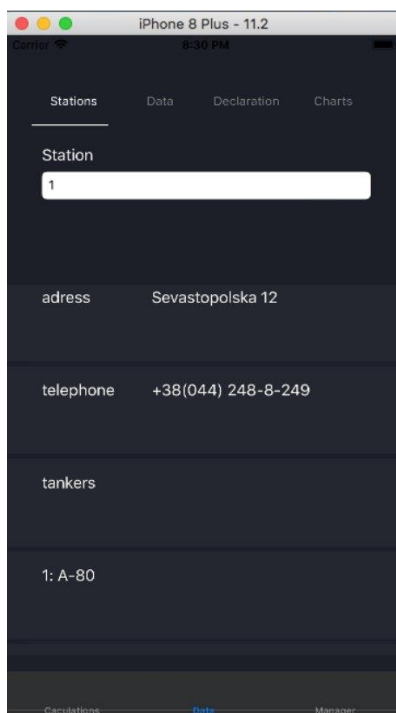


Рисунок 5.4. — Перегляд загальних даних про станцію АЗС

У вкладці Data є можливість переглянути дані про забруднення та виплати за них. Кожен запис представляє собою дані по викиди палива та виплати за них по кожному з можливих видів. Записи поділяються за часовим проміжком та приналежністю до певної станції.

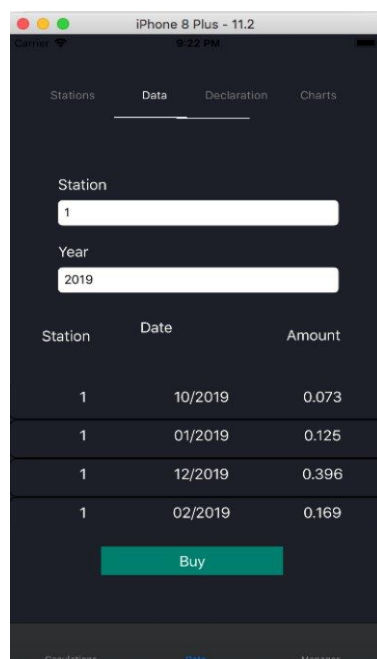
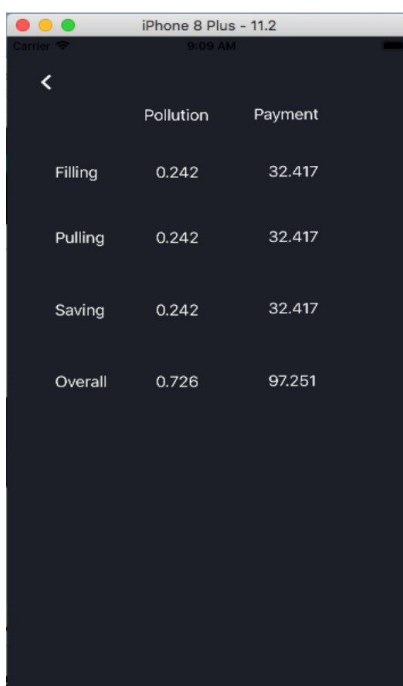


Рисунок 5.5 — Вивід збережених даних про викиди та виплати за них

При виборі конкретного запису, відбувається перехід на вікно, де буде відображена детальна інформація про викиди по типам та загальна кількість викидів за вказаними даними. В вікні є інформація зодо викидів при заповненні цистерни, заправці автомобіля, збереженні палива та загальний обсяг викидів зі вказаних джерел. За вказаними полями є інформація щодо об'єму викидів та виплат за них.



The screenshot shows a mobile application interface on an iPhone 8 Plus. At the top, the status bar displays 'Carrier', signal strength, 'iPhone 8 Plus - 11.2', and the time '9:09 AM'. Below the status bar is a dark navigation bar with a white back arrow icon on the left. The main content area is a table with a dark background and white text. The table has two columns: 'Pollution' and 'Payment'. The rows are 'Filling', 'Pulling', 'Saving', and 'Overall'. The 'Pollution' column contains values 0.242, 0.242, 0.242, and 0.726 respectively. The 'Payment' column contains values 32.417, 32.417, 32.417, and 97.251 respectively.

	Pollution	Payment
Filling	0.242	32.417
Pulling	0.242	32.417
Saving	0.242	32.417
Overall	0.726	97.251

Рисунок 5.6 — Вікно з детальною інформацією по викидам

У вкладці Declaration є можливість отримати дані для заповнення податкової декларації. За обраним роком можна побачити код забруднюючої речовини, об'єм викидів та виплати за них. Викиди та виплати рахуються щоквартально по нарастаючій, починаючи з першого кварталу календарного року.

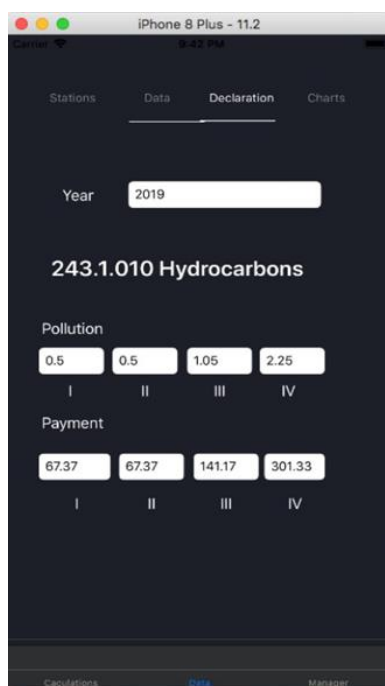


Рисунок 5.7 — Результат роботи класифікатора для тромбону

У вкладці Charts є можливість переглянути дані у графічному вигляді. Окрім опцій вибору року та індексу станції, користувач може обрати одне з можливих графічних представлень даних: викиди за поточний рік, динаміка росту викидів, максимальна потужність викиду з АЗС.

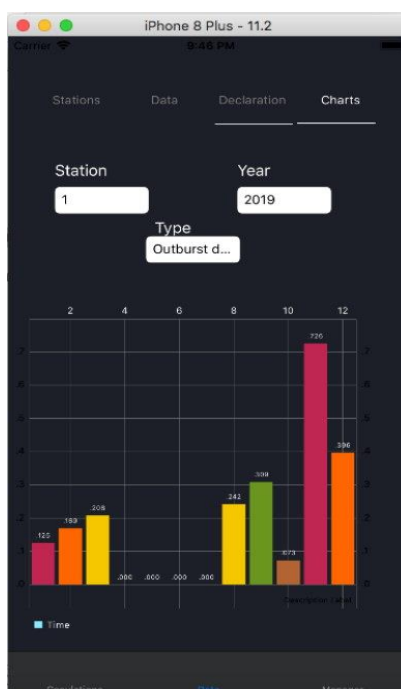


Рисунок 5.8 — Графічне представлення даних

5.4 Супровід додатку

При супроводі додатку є можливість додавати, видаляти, оновлювати склад мережі АЗС та видалення документів бази даних.

Вікно для створення нової АЗС передбачає наявність текстових полів введення даних, що представляють загальну інформацію про АЗС. Серед них є наступні поля: адреса, район, телефон, контактна персона. Також, на екрані присутня кнопка, що дозволяє перейти у вікно з можливістю додати склад цистерн даної АЗС.

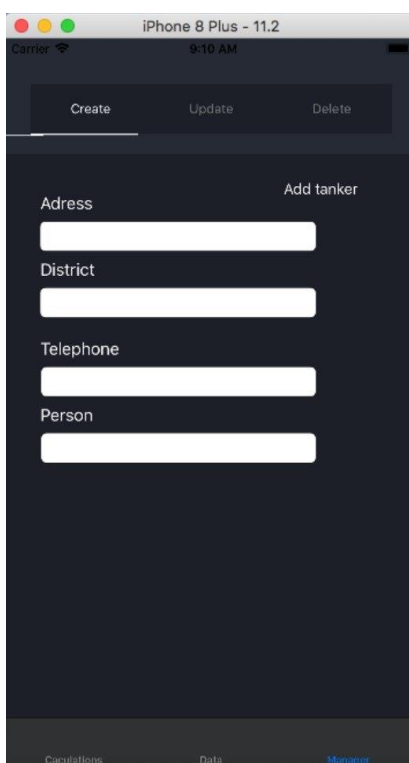


Рисунок 5.9 — Внесення нової станції до бази даних

При переході у вікно додавання цистерн, перед користувачем постають наступні поля: індекс цистерни та тип палива. Індекс генерується автоматично, тоді як тип палива обирається зі списку, що відображається при натисканні на текстове поле. Також, користувачу доступні дві кнопки – повернення на попередній екран та збереження інформації. Кнопка для повернення на попередній екран знаходиться у верхньому лівому куті екрану. У випадку, якщо користувач натисне на неї, увесь

обраний склад цистерн не буде збережено та дії користувача буде втрачено. Якщо користувач натисне кнопку Save, відбудеться запит до бази даних та введені раніше параметри буде збережено у вигляді нової станції мережі АЗС.

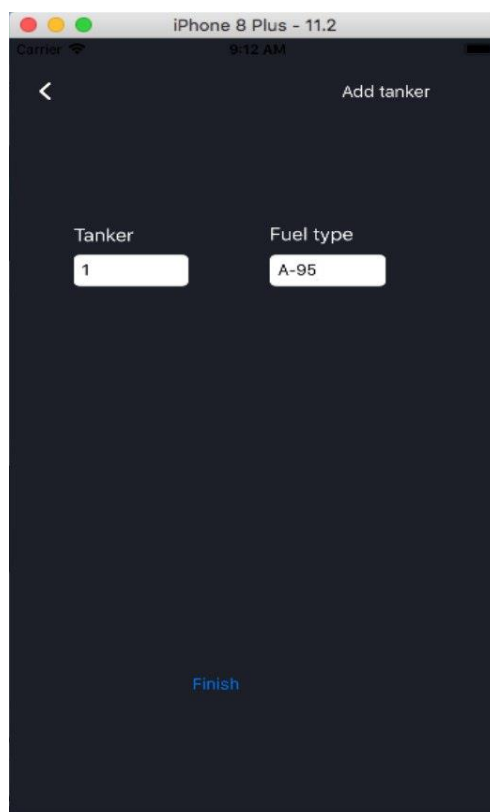


Рисунок 5.10 — Внесення нових цистерн

При переході на вікно видалення станцій, перед користувачем постає текстове поле, в якому можна обрати станцію. Після вибору індексу вибраної АЗС, в таблиці буди виведена інформація про обрану АЗС (рис. 5.11). Також, інтерфейс містить кнопку , з допомогою якої можна видалити обрану станцію. При натисканні буде виведено повідомлення, в якому користувачу потрібно буде підтвердити видалення станції. В випадку, якщо користувач обрав Yes, будуть виконані запити на видалення даних про обрану станцію та результатів її обчислень. Для перевірки здійснення операції, користувач може заново перевірити результат операції, натиснувши на текстове поле та перевіривши відсутність видаленого індексу.

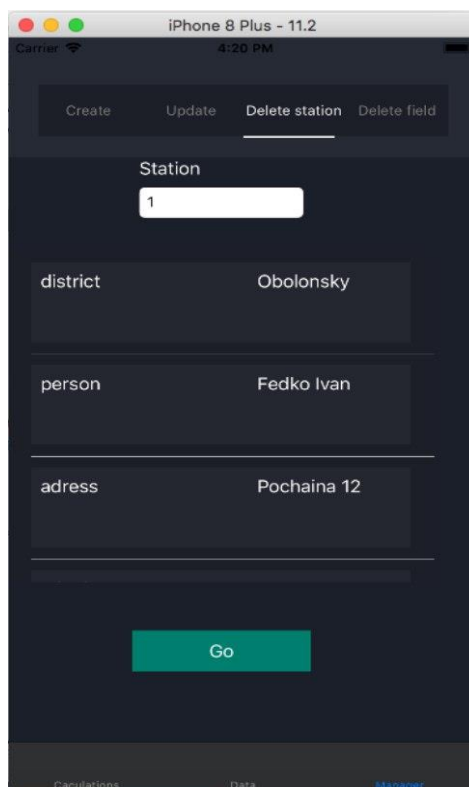


Рисунок 5.11 — Вікно видалення станції мережі

Перейшовши на вкладку видалення документів бази даних, перед користувачем постає екран з трьома текстовими пошлями та таблицею. Текстові поля дають можливість фільтрації даних за наступними параметрами: станція, часовий проміжок та тип викиду. Після здійснення фільтрації даних та здійснення запиту, в секціях таблиці будуть виведені усі документи, що відповідають введеним параметрам. Кожна секція містить наступні дані: станція, часовий проміжок та об'єм викиду.

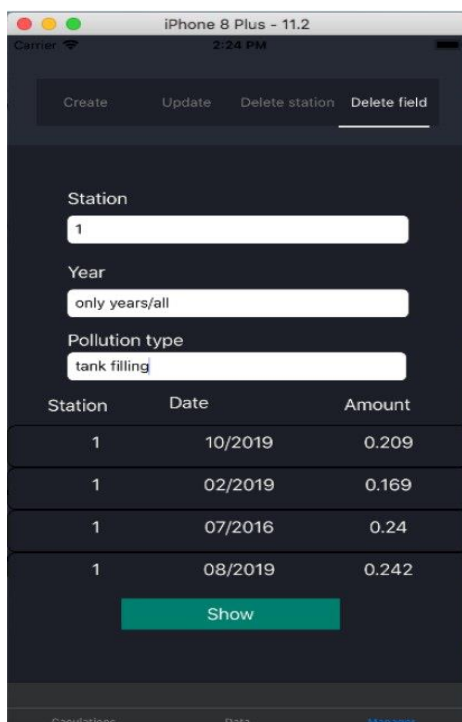


Рисунок 5.12 — Вікно станції мережі

Натиснувши на обрану секцію, відбувається перехід на екран детального перегляду документа. Після переходу на екран детального прешляду документу, перед користувачем постає таблиця з усіма полями, що містить обраний документ. До його складу входять: індекс станції, індекс танкера, часовий проміжок, тип палива, тип викиду, об'єм викиду та плата за даний об'єм викиду. На екрані присутні дві кнопки – видалення та повернення на попередній екран. Кнопка повернення знаходиться у верхньому лівому куті екрану. При натиску на неї, буде здійснено перехід на екран видалення полів. Якщо користувач натисне на кнопку видалення документу. При натисканні буде виведено повідомлення, в якому користувачу потрібно буде підтвердити видалення докумнету з бази даних. В випадку, якщо користувач обрав Yes, будуть виконані запити на видалення документа. Після здійснення обраної операції, буде здійснено перехід на попередній екран.

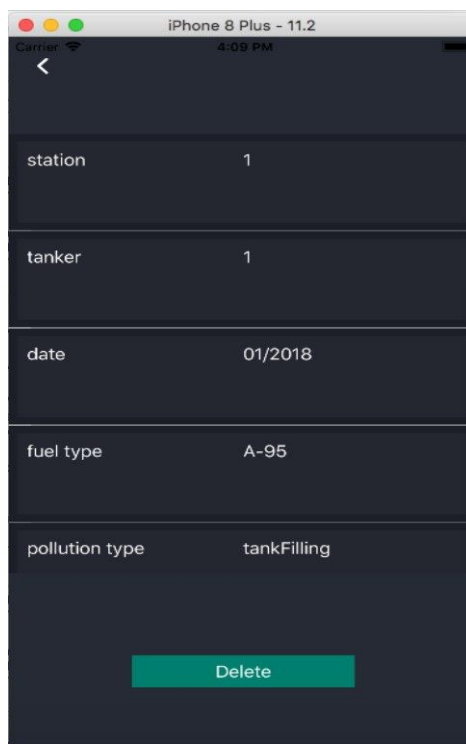


Рисунок 5.13 — Вікно з детальною інформацією документу

Висновки до розділу 5

Можливими напрямками розвитку додатку можна вважати наступні кроки:

- можливість аутентифікації та роботи з різними мережами АЗС;
- розширення податкової декларації;
- можливість запитів до мережі АЗС для редагування її складу;
- створення API з даними про обсяг використаного палива та його

подальше використання;

- адаптація мобільного додатку для роботи на операційній системі macOS;
- розширення даних, представлених в графічному вигляді.

6 РОЗРОБКА СТАРТАП-ПРОЕКТУ

В даному розділі буде розглянуто основні пункти стартап-проекту: опис ідеї проекту, технологічний аудит проекту, аналіз ринкових модливостей запуску проекту, розроблення ринкової стратегії проекту, розробка маркетингової ідеї проекту.

6.1 Опис ідеї проекту

В даному розділі будуть вказані опис ідеї стартап – проекту та розглянуті характеристики проекту. Нижче, в таблиці 6.1 вказана основна ідея проекту, розбито на пункти напрямки застосування та пояснені вигоди для користувача.

Таблиця 6.1 Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка модуля для обчислення екологічних викидів мережі АЗС	Обчислення викидів	Можливість обчислювати різні типи викидів
	Обчислення виплат	Можливість отримати необхідну інформацію для заповнення податкової декларації
	Аналіз отриманої інформації	Можливість представлення аналітичної інформації в графічному вигляді

Отже, провідними напрямками застосування: обчислення викидів, обчислення виплат, аналіз отриманої інформації.

В таблиці 6.2 перелічені техніко – економічні характеристики ідеї проекту. В наступних колонках йде порівняння проекту з аналогами та дається оцінка проекту по даному показнику в порівнянні з нижче зазначеними конкурентами.

Таблиця 6.2 Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко — економічні характери- стики ідеї	(потенційні) товари/концепції конкурентів				W (слаб ка сторо на	N (нейтр а льна сторон а)	S (сильна сторона)
		Мій проект	Конкур ент 1	Конку рент 2	Конкур ент 3			
1	Форма виконання	Додат ок	Додато к	Додат ок	Додато к		+	
2	Собівартіс ть	Низьк а	Середн я	Серед ня	Низька			+
3	Надійності	Висок а	Висока	Серед ня	Середн я			+
4	Наявність інтернету	Так	Ні	Так	Ні			+
5	Крос- платформ овий запуск програми	Ні	Ні	Ні	Ні	+		

Отже, серед описаних техніко – економічних характеристик ідеї, сильними

сторонами є собівартість, надійність та наявність інтернету. Нейтральною стороною є форма виконання, а слабкою стороною є відсутність кросс-платформового запуску програми.

6.2 Технологічний аудит ідеї проекту

В таблиці 6.3 проведено технологічний аудит ідеї проекту. В таблиці перелічені програмні інструменти, що описані за наступними характеристиками: ідея проекту, технології та реалізація, наявність технологій та доступність цих технологій. Після аналізу ідей проекту за вказаними складовими, буде зроблено висновок щодо обраних технологій реалізації проекту.

Таблиця 6.3 Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Графічне представлення даних	Charts	Наявна	Доступна
		SciCharts	Наявна	Платна, доступна
2	Використання хмарної СУБД	Cloud firestore	Наявна	Доступна
		Realtime Database	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: Charts, Cloud firestore				

Після розгляду ідей проекту та технологій її реалізації, було обрано безплатно бібліотеку Charts, так як серед усіх доступних інструментів для розробки графіків для

платформи iOS вона є безплатною, доступною та має задовільний функціонал. Firebase була обрана в якості платформи розробки мобільного додатку. Ця платформа дозволяє зберігати дані в хмарному сховищі, має інструменти для аутентифікації та авторизації користувача, та безплатна для невеликих проектів.

6.3 Аналіз ринкових можливостей запуску стартап-проекту

В таблиці 6.4 буде зроблена попередня характеристика стартап-проекту. Будуть перераховані основні показники стану ринку: кількість основних гравців, загальний обсяг продаж, якісна оцінка динаміки ринку, характер обмежень для входу на ринок, вимоги до стандартизації та сертифікації, середня норма рентабельності. вказані та описані їх характеристики.

Таблиця 6.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	15000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	30%

В таблиці 6.5 буде описана характеристика потенційних клієнтів стартап-проекту. В таблиці вказано потребу, що формує ринок, цільову аудиторію, відмінність у поведінці різних потенційних цільових груп клієнтів та вимоги споживачів до товару.

Таблиця 6.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінність у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Наявність небхідного функціоналу в одній програмі	Мережі АЗС	—	Зручний, ефективний та програмний продукт

При реалізації стартапу можливі ситуації, що можуть представляти загрозу для реалізації проекту. Необхідно передбачити та перерахувати загрози та запропонувати дії по їх вирішенню. В таблиці 6.6 будуть перераховані фактори загроз. Для кожного фактору буде описаний зміст загрози та можлива реакція компанії.

Таблиця 6.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Зміна законодавства	Відсутність функціоналу, що б відповідав новим вимогам	Оперативне оновлення програмного про
2	Недоступність платформи	Неможливість запустити програму на наявній техніці	Розробка аналогів на інші платформи

Отже, серед можливих загроз може бути зміна законодавства або недоступність платформи. Зміна законодавства є більшою загрозою, оскільки, при зміні вимог до оподаткування та обчислення викидів, функціонал програми не відповідатиме вимогам для вирішення поставлених задач. В наслідок недоступності платформи, користувач може відмовитись від використання програмного продукту. В подальшому будуть розроблені аналоги на інші платформи та подібна загроза буде ліквідована. Навіть при відсутності аналогів, запуск додатку можливий на емуляторах для платформи iOS.

При реалізації стартап-проекту будуть можливості, якими варто скористатись. В таблиці 6.7 будуть описані фактори можливостей, їх зміст та можлива реакція компанії.

Таблиця 6.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Реакція компанії
1	Збільшення попиту на програмний продукт	Збільшення аудиторії	Просування програмного продукту

Можливою реакцією на збільшення попиту на програмний продукт є залучення нової аудиторії шляхом просування програмного продукту.

Таблиця 6.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного Середовища	В чому проявляється дана Характеристика
Тип конкуренції	Монополістична
За рівнем конкурентної боротьби	Національний
За галузевою ознакою	Внутрішньогалузева
Конкуренція за видами товарів	Товарно-видова
За характером конкурентних переваг	Не цінова
За інтенсивністю	Не марочна

В таблиці 6.9, при огляді конкурентної ситуації на ринку, потрібно розглянути різні складові аналізу, а саме: прямі конкуренти в галузі, потенційні конкуренти, постачальники, клієнти, товари-замінники. Необхідно зробивши висновок, що характеризуватиме вказаний критерій.

Таблиця 6.9 Аналіз конкуренції в галузі за М. Портером

Складові Аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
------------------	---------------------------	-----------------------	---------------	---------	------------------

Продовження таблиці 6.9

	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
Висновки:	З конкуренти . 2 конкурент має найбільш подібний функціонал	Конкуренція , наявність готових рішень	Відсутні	Зручність використання	Вигідніша цінова пропозиція, покращений функціонал

Провівши аналіз конкуренції в галузі за М. Портером, можна виділити основні сильні сторони продукту, які б допомогли стати конкурентоспроможним на ринку – доступна ціна, зручність користувацького інтерфейсу, простота у використанні та швидкість, а також використання Firebase а якості сховища даних та сервера.

В таблиці 6.10 будуть вказані основні фактори конкурентоспроможності проекту та їх обґрунтування.

Таблиця 6.10 Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Доступна ціна	Програмний продукт доступний малим та починаючим підприємствам

Продовження таблиці 6.10

2	Функціонал	Розширений функціонал в порівнянні з іншими функціонал
3	Швидкість	Швидка та безпроблемна робота додатку
4	Зручність інтерфейсу	Інтуїтивно зрозумілий користувацький інтерфейс

За нижче вказаними факторами був проведений порівняльний аналіз сильних та слабких проекту. Була складена таблиця, де в кожній стрічці вказані фактори конкурентоспроможності. Було здійснене обґрунтування кожного фактору.

Таблиця 6.11. Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бади 1-20	Рейтинг товарів-конкурентів у порівнянні з						
			-3	-2	-1	0	1	2	3
1	Доступна ціна	18			+				
2	Функціонал	17			+				
3	Швидкість	15				+			
4	Зручність інтерфейсу	17			+				

В таблиці 6.12 наведено SWOT-аналіз стартап-проекту. Потрібно вказати сильні та слабкі сторони, можливості та загрози, що можуть виникнути при впровадженні проекту.

Таблиця 6.12. SWOT-аналіз стартап-проекту

Сильні сторони: доступна ціна, функціонал, швидкість, зручність інтерфейсу	Слабкі сторони: Обмеженість платформи
Можливості: обчислення викидів обчислення виплат аналіз інформації	Загрози: Зміна вимог до підприємств, поява нових конкурентів

Після проведення SWOT-аналізу, можна перерахувати сильні сторони проекту: доступна ціна, функціонал, швидкість, зручність інтерфейсу, серед слабких сторін можна назвати обмеженість реалізації продукту однією платформою.

Таблиця 6.13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива	Ймовірність отримання ресурсів	Строки реалізації
1.	Створення додатку на платформі iOS	75%	7 місяців
2.	Створення додатку на платформі Android	50%	10 місяців

6.4 Розроблення ринкової стратегії проекту

В таблиці 6.14 вказані групи цільових груп потенційних споживачів та описані за наступними характеристиками: опис профілю цільової групи, готовність споживачів сприйняти продукт, орієнтований попит в цільових сегментах, інтенсивність конкуренції та простота входу в сегмент. Потрібно здійснити висновок щодо вибору цільової групи.

Таблиця 6.14 Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Мережі АЗС	+	Середній	-	Нормальна
2	Інші підприємства	-	-	-	Нормальна
Які цільові групи обрано: мережі АЗС					

Отже, основною обраною цільовою групою було обрано мережі АЗС.

В таблиці 6.15 буде визначення базової стратегії розвитку. Крім самої базової стратегії розвитку, потрібно вказати обрану альтернативу розвитку проекту, стратегію охоплення ринку, ключові конкурентоспроможні позиції відповідно до обраної альтернативи.

Таблиця 6.15 Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку Проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної Альтернативи	Базова стратегія розвитку*
1	Створення додатку на платформі iOS	Спеціалізація	Доступна ціна, функціонал, швидкість, зручність інтерфейсу	Диференціація

Можна перерахувати наступні базові стратегії розвитку: обрана альтернатива розвитку проекту — створення додатку на платформі iOS, стратегія охоплення ринку — спеціалізація, ключові конкурентоспроможні позиції відповідно до обраної альтернативи — доступна ціна, функціонал, швидкість, зручність інтерфейсу.

В таблиці 6.16 буде розглянута стратегія конкурентної поведінки

Таблиця 6.16 Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1.	Ні	Забирати існуючих	Так	Зайняття конкурентної ніші

Отже, було визначено базову стратегію конкурентної поведінки — зайняття конкурентної ніші, адже програмний продукт буде націлений на один ринковий сегмент. Далі, в таблиці 6.17 буде визначення стратегії позиціонування

Таблиця 6.17 Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап проекту	Вибір асоціацій, які мають сформувати позицію проекту
----------	--	---------------------------------	--	---

Продовження таблиці 6.17

1	Доступна ціна, функціонал, швидкість, зручність інтерфейсу	Диференціація	Швидкість та особливості інтерфейсу дозволять провести обчислення та отримати доступ до інформації	Функціональність, зручність, доступність
---	---	---------------	---	--

Отже, було визначено стратегію позиціонування, а саме визначено основні вимоги до товару цільової аудиторії: доступна ціна, функціонал, швидкість, зручність інтерфейсу; базову стратегію розвитку: диференціація; ключові конкурентоспроможні позиції стартап-проекту: швидкість та особливості інтерфейсу дозволять провести обчислення та отримати доступ до інформації. Також сформовано комплексну позицію проекту: функціональність, зручність, доступність.

6.5 Розробка маркетингової програми

В таблиці 6.18 визначені ключові переваги концепції потенційного товару. Перераховані основні потреби, вигода, яку пропонує товар та ключові переваги перед конкурентами.

Таблиця 6.18 Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)

Продовження таблиці 6.18

1.	Доступна ціна	Вигідна цінова пропозиція	Ціна нижче середнього
2.	Функціонал	Функціонал покриває широкий спектр задач	Наявність широкого спектру рішень
3.	Швидкість	Швидка робота на усіх iOS девайсах	???Перевага в швидкості
4.	Зручність інтерфейсу	Відповідність iOS Human interface guidelines	Комфортний інтерфейс для роботи з додатком

В таблиці 6.19 буде розроблена трирівнева маркетингова модель товару. Є три рівні товару: товар за задумом, товар у реальному виконанні та товар із підкріпленням. На першому буде описана сутність та складові товару за задумом. На другому рівні потрібно описати його властивості, якість, маркування та компанію. На третьому рівні буде вказано інформацію про товар із прикріпленням до та після продажу. Після формування таблиці, слід вказати, яким чином потенційний товар буде захищено від копіювання.

Таблиця 6.19 Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Обчислення екологічних викидів та виплат, аналіз та представлення даних в одному програмному продукті

Продовження таблиці 6.19

II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Доступна ціна Функціонал Швидкість Зручність інтерфейсу	Нм	Тх
	Якість: Перевірка шляхом тестування		
	Маркування: Немає		
	Компанія: назва “А app”		
III. Товар із підкріпленням	До продажу: налаштування		
	Після: підтримка програмного продукту		
За рахунок чого потенційний товар буде захищено від копіювання: патент			

В табл. 6.20 будуть визначені та вказані межі встановлених цін. В таблиці потрібно вказати наступні дані: рівень цін на товаризамінники, рівень цін на товари-аналоги, рівень доходів цільової групи споживачів, верхня та нижня межі встановлення ціни на товар.

Таблиця 6.20 Визначення меж встановлення ціни

№ п/п	Рівень цін на товаризамінники	Рівень цін на товари- аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар
----------	----------------------------------	-------------------------------------	--	--

Продовження таблиці 6.20.

1.	15000	20000	425000	10000-25000 т. грн
----	-------	-------	--------	-----------------------

Для розповсюдження та продажу товару потрібно налагодити систему збуту, з допомогою якої клієнт зможе придбати вказаний товар. В таблиці 6.21 будуть перераховані параметри формування системи збуту, а саме: специфіка закупівельної поведінки цільових клієнтів, функції збуту, які має виконувати постачальник товару, глибина каналу збуту та оптимальна система збуту.

Таблиця 6.21 Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Підписка, покупка ліцензії???	Продаж	0(напрям),1(посередник)	Власна та через посередників

Останньою складовою маркетингової програми є розробка концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів. В табл 6.22 потрібно вказати: специфіка поведінки цільових клієнтів, канали комунікацій, якими користуються цільові клієнти, ключові позиції, обрані для позиціонування та завдання рекламного повідомлення.

Таблиця 6.22 Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення
1.	Завантаження з App Store	Інтернет	Доступна ціна, функціонал, швидкість, зручність інтерфейсу	Показати переваги додатку та зацікавити потенційну аудиторію

Отже, основним джерелом завантаження додатку буде App Store. Канало комунікації для цільових клієнтів – інтернет, ключові позиції позиціонування: доступна ціна, функціонал, швидкість, зручність інтерфейсу, завдання рекламного повідомлення — показати переваги додатку та зацікавити потенційну аудиторію.

Висновки до розділу 6

В розділі 6 було розглянуто основні пункти стартап-проекту: опис ідеї проекту, технологічний аудит проекту, аналіз ринкових модливостей запуску проекту, розроблення ринкової стратегії проекту, розробка маркетингової ідеї проекту. Стартап проект завершений та готовий до презентації.

ВИСНОВКИ

При виконанні роботи було виконано дослідження вимог та можливих методів обчислення викидів та виплат за них. Були розглянуті доступні програмні засоби та реалізовано програмний продукт. Все це допомогло створити зручний програмний засіб для вирішення ряду проблем, що стоять перед АЗС. Використання хмарної бази Firebase та бібліотеки Charts надали можливість запровадити єдиний інтерфейс для взаємодії з продуктом. Таким чином, були виконані наступні задачі:

- введення та обчислення викидів;
- представлення даних в графічному та текстовому форматах;
- аналіз отриманих даних;
- можливість супроводу системи;
- подання даних для заповнення додатку до податкової декларації.

Хоча, в подальшому можливі різні варіанти розвитку програмного продукту, проект готовий до використання .

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Франчук Г. М. Оцінювання забруднення ґрунтів нафтопродуктами внаслідок діяльності автозаправних станцій / Г. М. Франчук, М. М. Радомська / Вісник НАУ. – 2009. – №1(38). – С. 46–49.
- 2 Про внесення змін до Закону України "Про охорону атмосферного повітря" <https://ips.ligazakon.net/document/view/T012556>
- 3 Як правильно заповнити декларацію з екоподатку. [Електронний документ] . – Режим доступу: <http://ecolog-ua.com/news/yak-pravylnno-zapovnyty-deklaraciyu-z-ekopodatku>
- 4 Франчук Г. М. Аналіз даних про токсичність паливно-мастильних матеріалів для людини / Г. М. Франчук, М. М. Николяк // Вісник НАУ. – 2007. – №3–4(33). – С. 54–58. Боровков А. И. Компьютерный инжиниринг / А. И. Боровков. — СПб: Изд-во Политехн. ун-та, 2012. — 93 с.
- 5 Соколова Е. В. К оценке экологической опасности выбросов автозаправочных станций (АЗС) для воздушного бассейна городских комплексов / Е. В. Соколова ; Сев.-Кавказ. гос. тех. ун-т // Вестник Северо-Кавказского государственного технического университета. - 2012. – N 1 (30). - С. 64-68.
- 6 Шубов Л.Я. Проблема загрязнения окружающей среды от деятельности АЗС / Л.Я. Шубов //Экология и промышленность России. - 2005. - № 12. - С. 34-39.
- 7 Визначення обсягів викидів стаціонарними джерелами [Електронний документ] . – Режим доступу: <http://www.visnuk.com.ua/ua/news/id/3138>
- 8 Ус В. Swift. Основы разработки приложений под iOS / Василий Усов. — Москва: Вильямс, 2016, 896 с.
- 9 Objective-C Tutorial [Електронний ресурс] . – Режим доступу: https://www.tutorialspoint.com/objective_c/index.htm
- 10 Xcode - Начало работы с Xcode [Електронний ресурс] . – Режим доступу: <https://riptutorial.com/ru/xcode>

- 11 Increase your iOS development productivity with AppCode. [Электронный ресурс] . – Режим доступа: <https://blog.usejournal.com/increase-your-ios-development-productivity-with-appcode-3f230d825b30>
- 12 Портнякин И. Эффективные пользовательские интерфейсы / Иван Портнякин. — Санкт-Петербург: Лори, 2011. — 600 с.
- 13 Хорстманн К. Облачные СУБД. Вводный курс / Кей Хорстманн. — М: Вильямс, 2014. — 208 с.
- 14 Firebase databases [Электронный ресурс]. – Режим доступа: <https://firebase.google.com/docs/database>
- 15 Firebase Cloud Functions, (or running code on Firebase Servers!) [Электронный ресурс] . – Режим доступа: <https://javebratt.com/firebase-cloud-functions/>
- 16 Cloud firestore [Электронный ресурс] . – Режим доступа: <https://firebase.google.com/products/firestore/>
- 17 A powerful chart graph framework [Электронный ресурс] . – Режим доступа: <https://iosexample.com/a-powerful-chart-graph-framework/>
- 18 Easy to use and highly customizable charts library for iOS [Электронный ресурс] . – Режим доступа: <https://iosexample.com/easy-to-use-and-highly-customizable-charts-library-for-ios/>
- 19 Electric Mobile Studio [Электронный ресурс] . – Режим доступа: <https://techtade.com/ios-emulator/electric-mobile-studios>
- 20 MVC [Электронный ресурс] . – Режим доступа: <https://developer.apple.com/library/archive/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>
- 21 Microsoft Azure [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/azure/>
- 22 Microsoft launches Project Brainwave [Электронный ресурс]. – Режим доступа: <https://techcrunch.com/2018/05/07/microsoft-launches-project-brainwave-its-deep-learning-acceleration-platform/>
- 23 iOS Human Interface guidelines [Электронный ресурс]. – Режим доступа: <https://developer.apple.com/design/human-interface-guidelines/>

- 24 What is BaaS [Електронний ресурс]. – Режим доступу: <https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/>
- 25 More about Firebase Auth [Електронний ресурс]. – Режим доступу: https://pub.dev/packages/firebase_auth
- 26 Document-oriented and relational data [Електронний ресурс]. – Режим доступу: <https://developer.couchbase.com/documentation/server/3.x/developer/dev-guide-3.0/compare-docs-vs-relational.html>
- 27 Манульное тестирование ПЗ. [Електронний ресурс]. – Режим доступу: <https://qalight.com.ua/baza-znaniy/ruchnoe-i-avtomatizirovannoe/>
- 28 Розроблення стартап-проекту [Електронний ресурс] : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016.
- 29 Lets talk about startup [Електронний ресурс]. – Режим доступу: <https://ktovdele.com/chto-takoe-startap.html>
- 30 Стартап-проект. Рекомендації до виконання розділу магістерської дисертації «Розроблення стартап-проекту» [Електронний ресурс]. – Режим доступу: <https://ela.kpi.ua/handle/123456789/27914>

ДОДАТОК А

Система розрахунку екологічних платежів мережі АЗС

Акт впровадження

УКР.НТУУ"КПІ ім. Ігоря Сікорського" _ТЕФ_АПЕПС_ТВ4197_19М

Сторінок 3

Київ 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали XVII Міжнародної
науково-практичної конференції
молодих вчених та студентів
м. Київ, 23-26 квітня 2019 року,

ТОМ 2



Київ- 2019

МОСКАЛЕНКО Ю.В., аспірант Машинне навчання для розв'язання логічних головоломок.	93
БАРАНЧЕНКО О.М., магістрант гр. ТВ-71мн Карієних - доц., к.т.н. Шатовалова С.І. Веб-сереловище для моделювання процесів міжагентної взаємодії в мережах Smart Grid.	94
ШВАЙКА Д.А., магістрант гр. ТР-81мн Карієних - доц., к.ф.-м.н. Тарнавський Ю.А. Використання техніки Structure from Motion в системі навігації.	95
ХАРАБАР В.В., магістрант гр. ТВ-81мн Карієних - доц., к.т.н. Газарін О.О. Система оцінювання екологічних збитків у мережі АЗС.	96
ОЛЕКСИЙ А.О., магістрант гр. ТВ-82 Карієних - доц., к.т.н. Газарін О.О. Інтелектуальна система розпізнавання та передбачення намірів користувача.	97
МЕЛЬНИЧЕНКО А.В., магістрант гр. ТВ-81мн Карієних - ст. викл., к.т.н. Шалденко О.В. Проблема формування схеми замкнутого простору у системах внутрішньої навігації.	98
МАРУНЯ А.В., магістрант гр. ТВ-81мн Карієних - доц., к.т.н. Газарін О.О. Застосування нейронних мереж в мобільних застосунках.	99
МАРИЧ Т.І., магістрант гр. ТВ-81мн Карієних - доц., к.т.н. Шатовалова С.І. Генерація елементів цифрового контенту на основі аналізу тексту.	100
КРЮЧКОВСЬКА А.В., магістрант гр. ТВ-81мн Карієних - ст. викл., к.т.н. Шалденко О.В. Програмний інструментарій вивчення заданих об'єктів на зображенні.	101
КРУТЛИК Д.С., магістрант гр. ТВ-81мн Карієних - доц., к.т.н. Шатовалова С.І. Система розпізнавання жестів рук для людинно-машинної взаємодії.	102
КОЖИНА Н.С., магістрант гр. ТВ-81мн Карієних - ст. викл., к.т.н. Шалденко О.В. Проблема вибору раціонального методу позиціювання користувача для системи навігації.	103
ЗАРИПЬКИЙ В.П., магістрант гр. ТВ-81мн Карієних - доц., к.т.н. Газарін О.О. Нейромережеве архітектурне рішення для обробки аудіосигналів.	104
ВИТВИЦЬКИЙ Д.А., магістрант гр. ТВ-81мн Карієних - ст. викл., к.т.н. Мажара О.О. Побудова сучасного веб-серверу на основі безсерверних технологій.	105
БРУНЬКО П.В., магістрант гр. ТВ-81мн Карієних - доц., к.т.н. Шатовалова С.І. Сегментація бур'янів на зображеннях з відеокамери наземного робота.	106
СОФІЄНКО А.Ю., студент гр. ТР-52 Карієних - доц., к.т.н. Шатовалова С.І. Серверна частина системи функціонування реєстру інформаційних ресурсів.	107
СОЛОМКИН Д.Г., студент гр. ЗП-ЗП-63 Карієних - ст. викл. Гайдаржи В.І.	

УДК 621.43.056:632.18

Магістрант 5 курсу, гр. ТВ-82мн Олексій А.О.
Доп., к.т.н. Гагарін О.О.

СИСТЕМА ОЦІНЮВАННЯ ЕКОЛОГІЧНИХ ЗБИТКІВ У МЕРЕЖІ АЗС

Нещодавні дослідження показали, що людська діяльність в 170 раз пришвидшує зміни клімату. Особливої уваги заслуговують викиди в атмосферу з АЗС. За радянських часів, АЗС будувались за межею значних населених пунктів та їх кількість регулювалась. Проте, з комерціалізацією автозаправної справи та збільшенням попиту на нафтопродукти, проблема почала набувати значних масштабів.

Викиди з АЗС поділяються на п'ять видів: викиди при зливні палива, викиди при випаровуванні палива, викид холостого пробігу, викид при зберіганні палива, викиди при заправці паливом. Подібні явища несуть загрозу навколишньому середовищу, що супроводжується накладенням грошового збору з боку держави.

Для вирішення цієї проблеми потрібні зручні інструменти, що відповідають викликам сучасності. Для цього був розроблений мобільний додаток для обчислень екологічних платежів за збитки, які можуть виникнути у АЗС мережі.

Даний програмний продукт може використовуватися не тільки підприємствами які володіють мережею АЗС, а й екологічними установами при проведенні екологічного моніторингу.

Функціонал додатку:

- можливість перегляду основних даних моніторингу на карті.
- реєстрація користувача.
- аутентифікація користувача.
- обчислення збитків за різними критеріями.
- обчислення екологічних виплат за збиток
- реалізація зручного доступу до даних для перегляду, аналізу та редагування

Розробка програмного продукту була здійснена на мові програмування Swift.

Основне середовище розробки – XCode 9.0.

Реалізація карт була проведена за допомогою Google Maps SDK.

Створення та редагування баз даних були здійснені за допомогою хмарної системи управління базами даних Cloud Firestore.

Перелік посилань:

1. Про затвердження Інструкції про порядок обчислення та сплати збору забруднення навколишнього природного середовища [Електронний ресурс]. – 2011. – Режим доступу до ресурсу: <http://zakon5.rada.gov.ua/laws/show/z0544-99>
2. Івасенко В.М. Вплив промисловості на стан атмосферного повітря в місті Києві [Текст] / В.М. Івасенко // Погляд у майбутнє приладобудування: IV наук.- техн. конф., 12 квітня 2011р. : тези доп. – Київ, 2011. – С.208.